Introducción a Apache Hadoop.

Caso práctico

Tim Bray (CC BY-SA)

Estamos en el año 2002, Doug y Mike trabajan en el proyecto Apache Nutch, un motor de búsqueda que pretende indexar mil millones de páginas web.

Después de analizar la arquitectura que han diseñado, llegan a la conclusión de que para poder cumplir su objetivo, y con la tecnología del momento, costaría más de medio millón de euros en hardware, y otro medio millón al año de costes de mantenimiento o ejecución.

¡No puede ser! ¡Esto es inviable! ¡Tenemos que darle una vuelta a la arquitectura!

Doug y Mike empiezan a buscar soluciones en el mercado que les permitan, por un lado, almacenar un volumen de datos enorme, y por otro, poder procesar todos los datos para generar los índices de búsqueda. Por si fuera poco, la solución debe ser económica, así que no es tarea sencilla.

Por suerte, descubren algo que solucionará su problema. Con el tiempo, lo que acaban de descubrir se llamará Apache Hadoop.

A lo largo de esta unidad vas a aprender por qué surge Apache Hadoop, cómo ha evolucionando y en qué estado se encuentra actualmente. Además, vas a poder entender cómo funciona, qué arquitectura tiene y qué funcionalidades ofrece, y sobre todo, entenderás qué beneficios aporta, así como las desventajas que tiene frente a otras tecnologías o las dificultades que tiene Hadoop para ser utilizado por las organizaciones.

Los contenidos de la unidad serán los siguientes:

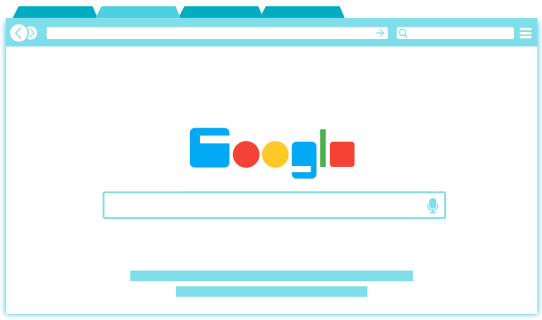
- Motivación, origen e historia de Apache Hadoop: te mostrará por qué se originó Hadoop, en qué se basó y cómo ha ido evolucionando desde su origen.
- ¿Qué es Apache Hadoop?: detallará qué es Hadoop y qué características principales tiene.

- Ecosistema y distribuciones: Hadoop es una plataforma con un ecosistema de herramientas asociado. Este apartado te mostrará las herramientas de este ecosistema y cómo se han agrupado dichas herramientas para facilitar su uso.
- Arquitectura de Hadoop: en este apartado conocerás dónde se instala, qué tipo de hardware necesita y cómo funciona.
- Beneficios, desventajas y dificultades: Hadoop tiene unos beneficios importantes, pero como toda nueva tecnología, tiene una serie de dificultades que se detallarán en este apartado.

1.- Motivación y origen.

Caso práctico

Doug y Mike están buscando una solución que permita almacenar mil millones de páginas y procesarlas para el proyecto Apache Nutch, un motor de búsqueda, como los que ya tienen Google o Yahoo. Aunque en el año 2002 no era habitual enfrentarse a problemas que requieran tratar con un volumen de datos tan grande, ya había compañías, como las que acabamos de mencionar, que habían resuelto ese problema.



200degrees (Dominio público)

¿Por qué no investigar entonces cómo Google o Yahoo han solucionado este reto?

En octubre del año 2003, Google publicó un paper sobre su Sistema de almacenamiento escalable denominado The **Google File System**. En este paper, Google explicó cómo resolvían la problemática de almacenar un gran volumen de datos (petabytes) a un bajo coste utilizando un modelo de almacenamiento distribuido.

Este paper sirvió de inspiración a Doug Cutting y Mike Cafarella para diseñar la solución de almacenamiento de Apache Nutch. ¡Sólo quedaba por resolver cómo procesar los datos para generar los índices de búsqueda!

En diciembre de 2004, Google publicó otro paper, cuyo título era **Map Reduce: Simplified Data Processing on Large Clusters**, donde se describía cómo había resuelto Google el procesamiento sobre conjuntos de datos voluminosos utilizando un paradigma de computación ya existente, MapReduce.

Este segundo paper resolvía, por lo tanto, el procesamiento de los datos para generar los índices de búsqueda de Apache Nutch.

Los papers de Google describían la solución de Google a alto nivel, pero no daban detalles sobre cómo se había implementado a nivel de código. Doug Cutting y Mike Cafarella empezaron a trabajar en la implementación dentro del proyecto Apache Nutch. Sin embargo, se encontraron con la dificultad de que el desarrollo era complejo, y que sólo dos personas necesitarían mucho tiempo para implementarlo. Además, se dieron cuenta de que Apache Nutch no podría sacar toda la potencia de lo que estaban desarrollando por limitaciones técnicas.

Por estos motivos, en 2006, Doug Cutting se incorpora a Yahoo! con el objetivo de aprovechar la capacidad y el equipo que esta compañía tenía, para poder implementar GFS

y MapReduce y ofrecerlo al mundo como código libre. Doug Cutting estaba convencido de que un sistema de almacenamiento masivo y económico, junto con un sistema de procesamiento de datos para grandes volúmenes de datos a bajo coste, iba a revolucionar el estado de la tecnología del momento, y por ese motivo, sacó GFS y MapReduce del proyecto original Apache Nutch y les dió entidad propia. Este nuevo proyecto se llamaría **Hadoop**.

Para saber más

Apache Software Foundation ((Apache License, Version 2.0)

Doug Cutting eligió el nombre Hadoop porque era el nombre de un peluche de elefante que tenía su hijo. El nombre le parecía fácil de pronunciar y de recordar, además de tener un componente emocional para él.

En 2007, Hadoop ya se había implementado en Yahoo! en una primera versión beta, y fue probado en una infraestructura de más de 1000 nodos.

En enero de 2008, Yahoo! donó Hadoop a Apache Software Foundation, pasando a ser un proyecto Top-Level de la fundación. A partir de este momento, una gran cantidad de compañías y de desarrolladores mostraron interés en Hadoop y comenzaron a contribuir en el proyecto, haciendo que éste creciera en gran medida en los siguientes años.

En el año 2009, Doug Cutting abandonó Yahoo! y se incorporó a Cloudera, que pretendía hacer de Hadoop un producto de uso empresarial, ya que hasta ahora había tenido una vocación más de investigación o innovación.

La primera versión estable de Hadoop, la 1.0, fue liberada en diciembre de 2011.

Para saber más

Si quieres echar un vistazo a los papers que Google publicó sobre Google File System y Map Reduce, aquí tienes los enlaces a las publicaciones:

- Google File System.
- MapReduce: Simplified Data Processing on Large Clusters

Autoevaluación

¿En qué año se puede decir que se originó Hadoop?

- ○ 2003
- ○ 2004
- ○ 2006
- ○ 2011

Incorrecto: en el 2003 fue publicado el paper de Google File System, en el que se fijaron Doug Cutting y Mike Cafarella para implementar un sistema de almacenamiento para Apache Nutch, pero todavía no existía Hadoop como proyecto.

Incorrecto: en el 2004 fue publicado el paper de MapReduce, en el que se fijaron Doug Cutting y Mike Cafarella para implementar un sistema de procesamiento de datos para Apache Nutch, pero todavía no existía Hadoop como proyecto.

Correcto

Incorrecto: en 2011 fue liberada la versión 1.0, pero Hadoop ya tenía unos años de existencia como proyecto en fases beta.

Solución

1. Incorrecto
2. Incorrecto
3. Opción correcta
4. Incorrecto

2.- Apache Hadoop a alto nivel.

Caso práctico

Roberto, el CTO de una gran empresa de transportes, tiene que implementar un sistema con el que poder analizar los datos de los GPS de todos los camiones para optimizar rutas, detectar comportamientos incorrectos de los conductores, o para poder predecir qué camiones deberían tener un mantenimiento porque hay una probabilidad elevada de que tengan un fallo en breve.

StartupStockPhotos (Dominio público)

Su amiga María, que es una ingeniera de datos en una empresa tecnológica, le ha hablado de Hadoop, diciéndole que es lo que debe utilizar.

Roberto necesita entender qué es Hadoop antes de dar un primer paso.

Hadoop es una plataforma que permite almacenar y procesar grandes volumenes de datos. No es una plataforma sencilla porque tiene muchos componentes, y además, se instala en muchos servidores.

Vamos a descubrir Hadoop empezando por un alto nivel, entendiendo qué es, para qué sirve y cómo funciona, y en los siguientes módulos iremos desgranando todos los componentes que pertenecen a la plataforma.

Para saber más

Apache Hadoop tiene su propia documentación oficial, donde puedes encontrar todos los detalles sobre la plataforma. Es una documentación extensa, pero si quieres acceder y ver su contenido, o para profundizar en algún punto, aquí tienes un enlace a la página oficial.

2.1.- ¿Qué es Apache Hadoop?

Apache Software Foundation (Apache License, Version 2.0)

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**.

Vamos a desgranar la definición, extrayendo toda la información que contiene:

Plataforma

Hadoop es una plataforma, lo que significa que es la base sobre la que construir aplicaciones. Se podría hacer el símil a que Hadoop es una caja de herramientas que proporciona un conjunto de herramientas con las que construir una gran variedad de aplicaciones que requieran almacenar y procesar grandes volúmenes de datos. La selección de qué herramienta utilizar para cada aplicación la realizaremos en función de las necesidades de cada caso de uso.

Otras soluciones, como MongoDB u otras bases de datos NoSQL no se consideran plataformas, ya que tienen un único propósito y ofrecen un tipo de funcionalidad.

Íñigo Sanz (Dominio público)

Opensource

Hadoop no es una solución comercial, sino que todo su código es libre y por lo tanto, no hay que pagar licencias o costes de adquisición del software de la plataforma.

Almacenar

Hadoop ofrece la capacidad de almacenar y recuperar datos mediante un sistema de ficheros que se llama HDFS, que veremos más adelante.

Procesar

Hadoop ofrece la capacidad de crear aplicaciones que procesen los datos almacenados en el sistema de ficheros, tanto de forma batch como real-time.

Coste

El coste de implantar una plataforma Hadoop es órdenes de magnitud más bajo que otras soluciones tradicionales de almacenamiento y procesamiento de datos, como podrían ser las bases de datos relacionales o los sistemas mainframe.

Volumen

Permite almacenar prácticamente cualquier volumen de datos, desde volumetrías pequeñas (megabytes) a volumetrías muy altas (petabytes).

Estructura

Los datos que pueden almacenarse y procesarse en Hadoop pueden tener cualquier tipo: estructurados, semiestructurados o datos no estructurados.

Íñigo Sanz (Dominio público)

Distribuido

Hadoop se basa en una infraestructura que tiene muchos servidores (también llamados nodos) que trabajan conjuntamente para almacenar o para procesar los datos, a diferencia de los sistemas centralizados, donde todo se realiza en un único servidor.

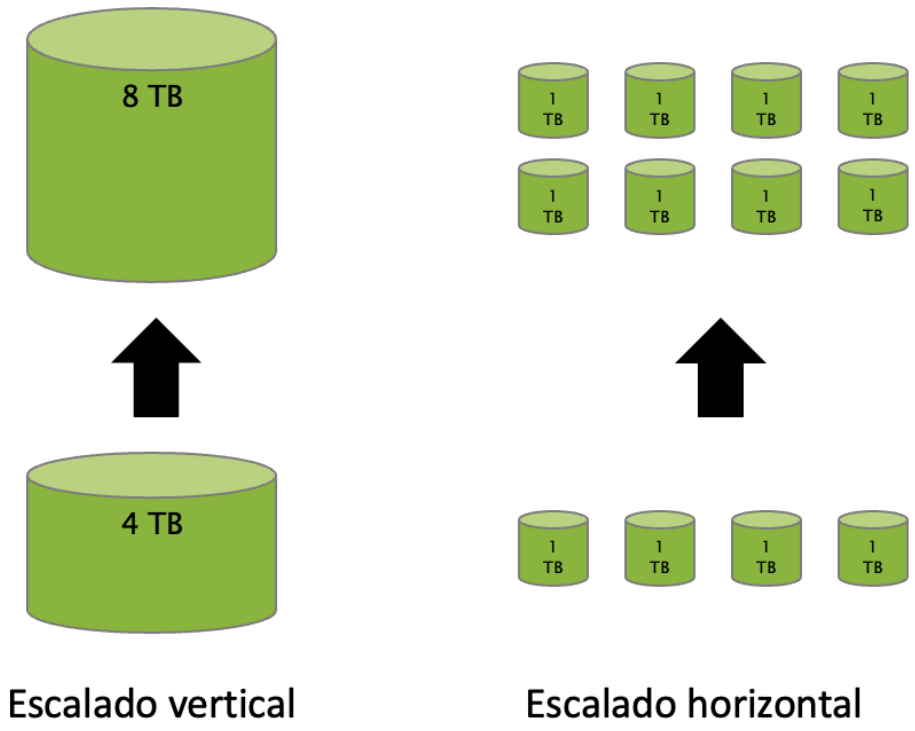
Escalable

Hadoop permite crecer en infraestructura (servidores) hasta adecuarse a las necesidades de almacenamiento o procesamiento del caso de uso. El modelo de escalado es horizontal, es

decir, si nuestra plataforma necesita crecer, en lugar de cambiar el servidor por uno de mayor capacidad (escalado vertical), se añaden más servidores del mismo tipo. Este tipo de escalabilidad tiene dos ventajas principales:

1.- El coste de incrementar la capacidad es **lineal**, es decir, si mi plataforma tiene una capacidad de 1 petabyte, y necesitamos incrementarla a 2 petabytes, el coste será el doble del coste inicial que hubo. En los sistemas de escalado vertical, el coste suele ser exponencial, es decir, incrementar de 1 terabyte a 2 terabytes puede suponer un coste 3 veces superior. Por ejemplo, echa un vistazo al coste de un disco duro de 3 terabytes y a un disco duro de 6 terabytes, ¿a que el disco de 6 terabytes cuesta mucho más que dos discos de 3 terabytes?

2.- La capacidad máxima vendrá determinada por el número máximo de servidores que se puede añadir. En el caso de Hadoop, el límite está en torno a unos 10.000 nodos (un nodo puede almacenar muchos terabytes). En los sistemas que tienen escalado vertical, el límite de crecimiento lo determina el tamaño máximo de un servidor que se puede comprar.



Íñigo Sanz (Dominio público)

Tolerante a fallos

Hadoop es una plataforma que garantiza que ante la caída de uno de los servidores, el sistema sigue funcionando y no se pierden datos. Esto es fundamental por varios motivos:

- Un sistema empresarial, es decir, un sistema que se va a utilizar para dar soporte a las operaciones de una empresa, debe funcionar correctamente 24x7, ¿a quién le gusta que cuando quiere hacer una llamada de teléfono o una transferencia bancaria le digan que en ese momento no se puede realizar la operación? O peor, ¿a quién le gustaría que el banco te comunique que ha perdido tus datos y que no sabe cuánto dinero tienes en la cuenta?
- Al tener un modelo distribuido, y estar formado por múltiples servidores, la probabilidad de que un servidor se rompa cada día es muy elevada. Piensa en una plataforma

Hadoop de 1.000 servidores, con 12 discos duros por servidor, es decir, un sistema con 12.000 discos. Si un disco tiene una duración estimada de 3 años, significaría que todos los días se romperían 11 discos. Hadoop garantiza que aunque se rompan 11 discos cada día, no se van a perder datos y las aplicaciones seguirán funcionando correctamente.

Hardware commodity

Hadoop no requiere servidores específicos con unas exigencias muy concretas. El concepto de hardware commodity ya se ha tratado con anterioridad. Ojo, ¡no significa que Hadoop se puede desplegar con los portátiles reciclados de una oficina!

Virginia Department of Education. *Ejemplo de un despliegue de Hadoop en una universidad* (CC BY)

Acercamiento del procesamiento a los datos

Los sistemas de procesamiento masivo de datos tradicionales se basaban en tener un sistema de almacenamiento separado del sistema de procesamiento. Este modelo requiere que antes de hacer cualquier proceso, hay que leer todos los datos y transportarlos al sistema de computación. Este transporte se realiza por la red de comunicaciones, que no tiene un ancho de banda comparable con el ancho de banda de lectura en disco y procesamiento en la CPU de la misma máquina. En los sistemas tradicionales, el cuello de botella siempre es la red de comunicaciones.

Debes conocer

Hadoop no fue la primera plataforma capaz de almacenar y procesar un volumen de datos grande.

Haz la siguiente reflexión: ¿los bancos, en los años 90, necesitaban gestionar un volumen de datos grande? Piensa en la cantidad de transferencias, pagos con tarjeta, movimiento de la valoración de las acciones cada día, etc. que un banco tiene que almacenar y tratar.

Efectivamente, ya había casos de uso "Big Data" antes de la aparición de Hadoop u otras tecnologías Big Data, y por ejemplo, los bancos utilizaron sistemas mainframe para almacenar todos los datos de su operativa así como su procesamiento. La principal diferencia entre Hadoop y otros sistemas tradicionales es que el coste es órdenes de magnitud inferior (10 veces, 1.000 veces, ...).

Autoevaluación

¿Cuál de las siguientes afirmaciones sobre Hadoop es falsa?

- [ ] Si quiero instalar y usar Hadoop, no tengo que pagar un coste de licencia.
- [ ] Hadoop permite almacenar ficheros de vídeo y procesarlos.
- [ ] El coste de instalar y operar una plataforma Hadoop es más o menos similar al de una base de datos relacional tradicional (Oracle, IBM DB2, etc.)
- [ ] Hadoop tiene una gran capacidad de almacenamiento, pero está limitado en cuanto a la capacidad de procesamiento
- [ ] Hadoop requiere servidores muy específicos, con al menos 1 terabyte de memoria RAM

Mostrar retroalimentación

Solución

1. Incorrecto
2. Incorrecto
3. Correcto
4. Correcto
5. Correcto

2.2.- Ecosistema Hadoop y distribuciones.

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	Logotipo
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 Software Foundation (Apache License)
Apache Pig	Utilidad para definir flujos de datos de transformación o consulta mediante un lenguaje de scripting.	 Apache Pig Apache Software Foundation (Apache License)
Apache HBase	Base de datos NoSQL de tipo columnar que permite el acceso aleatorio, atómico y con operaciones de edición de datos.	 Apache Software Foundation (Apache License)

Apache Flume	Componente para ingestar streams de datos procedentes de sistemas real-time en Hadoop.	Apache Software Foundation (Apache License)
Apache Sqoop	Componente para importar o exportar datos estructurados desde bases de datos relacionales a Hadoop y viceversa.	Apache Software Foundation (Apache License)
Apache Oozie	Herramienta que permite definir flujos de trabajo en Hadoop así como su orquestación y planificación.	Apache Software Foundation (Apache License)
Apache ZooKeeper	Herramienta técnica que permite sincronizar el estado de los diferentes servicios distribuidos de Hadoop.	Apache Software Foundation (Apache License)
Apache Storm	Sistema de procesamiento real-time de eventos con baja latencia.	Apache Software Foundation (Apache License)
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 Software Foundation (Apache License)
Apache Kafka	Sistema de mensajería que permite recoger eventos en tiempo real así como su procesamiento.	Apache Software Foundation (Apache License)
Apache Atlas	Herramienta de gobierno de datos de Hadoop.	Apache Software Foundation (Apache License)
Apache Accumulo	Base de datos NoSQL que ofrece funcionalidades de acceso aleatorio y atómico.	Apache Software Foundation (Apache License)
Apache Mahout	Conjunto de librerías para desarrollo y ejecución de modelos de machine learning utilizando las capacidades de computación de Hadoop.	Apache Software Foundation (Apache License)
Apache Phoenix	Capa que permite acceder a los datos de HBase mediante interfaz SQL.	Apache Software Foundation (Apache License)
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 Software Foundation (Apache License)

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).	 ™ Apache Software Foundation (Apache License)

Recomendación

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.

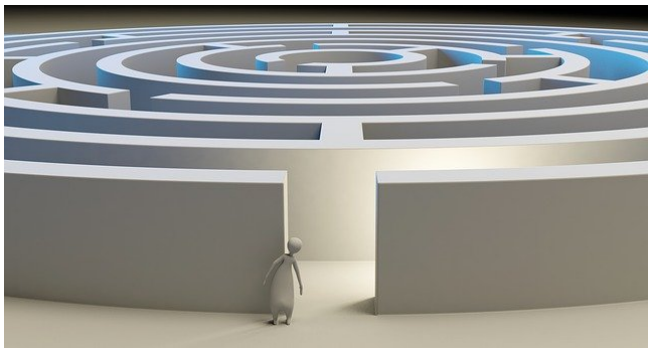
Los más utilizados son: Apache Spark, Apache Hive y Apache Kafka, además de los componentes core: HDFS y YARN.

Cada componente es un proyecto Apache independiente, lo que impacta, entre otros a:

- **Política de versionado (periodicidad, identificación, …)**: cada componente tiene sigue su propio camino en cuanto a cuándo se publican las nuevas versiones, qué mejoras o evoluciones incluyen, etc.
- **Dependencias del proyecto con otras versiones de componentes del ecosistema y librerías externas**: los componentes suelen tener dependencias entre ellos. Por ejemplo, Hive tiene dependencia de HDFS, o Phoenix de HBase. Las dependencias suelen ser difíciles de gestionar, por ejemplo, porque una versión de Phoenix requiere una versión específica de HBase.
- **Roadmap y estrategia del proyecto**: al tener grupos de trabajo diferentes, cada proyecto tiene su propia estrategia en cuanto a cómo evolucionar la solución, cuándo adaptarse a cambios externos, etc. y no siempre están alineados.
- **Commiters / desarrolladores**: los desarrolladores de cada proyecto son diferentes.

Por este motivo, realizar una **instalación** de toda una plataforma Hadoop con sus componentes asociados de forma independiente (lo que se denomina Hadoop Vanila) resulta

muy complicado. Por ejemplo, al instalar la versión X de Phoenix necesitas la versión Y de HBase, pero otro componente (Hive, por ejemplo), requiere la versión Z de HBase.



qimono (Dominio público)

La misma dificultad ocurre para la **resolución de incidencia**s que puedan ocurrir en la plataforma cuando se ejecuta en producción. Es decir, si has conseguido poder instalar todos los componentes y que no hayan fallos de configuración o dependencias entre ellos (¡enhorabuena!), es bastante normal que puedan ocurrir errores cuando la plataforma está usándose en producción (algún servidor se queda sin espacio en disco, por ejemplo, o simplemente hay un fallo al ejecutar un determinado trabajo). Si la plataforma se está ejecutando en producción y está dando soporte a las operaciones de una empresa, el fallo debe corregirse lo antes posible, y no es tarea fácil porque hay que buscar en muchos componentes para averiguar dónde puede estar el fallo.

Peggy und Marco Lachmann-Anke (Dominio público)

Para solventar las dos dificultades mencionadas, 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.
- **MAPR**: rehizo la mayor parte de componentes utilizando los mismos interfaces pero reimplementando el core para ofrecer un mayor rendimiento.

Debes conocer

En octubre de 2018 se anunció la fusión de las dos principales distribuciones, Cloudera y Hortonworks, que se hizo efectiva durante 2019, lo que generó un movimiento sin precedentes en el mercado de las tecnologías Big Data.

Asimismo, en 2019, MAPR, ante la imposibilidad para obtener fondos que financiaran su actividad, cesó su actividad, vendiendo todo su portfolio a HPE.

Con estos cambios, la única distribución de Hadoop existente hoy en día es la de Cloudera, que integra las funcionalidades relevantes de Hortonworks.

Reflexiona

¿Por qué crees que en un mercado que se supone que tiene suficiente volumen con el auge de Big Data o la Inteligencia Artificial, sólo hay una distribución Hadoop disponible?

Mostrar retroalimentación

La respuesta está en el auge del cloud.

Los principales proveedores de cloud, es decir, Amazon (AWS), Microsoft (Azure) y Google (Google Cloud), han lanzado y potenciado servicios de Hadoop gestionados, es decir, ofrecen la posibilidad de desplegar plataformas Hadoop en modalidad pago por uso.

La mayor parte de las empresas están inmersas en procesos de migración a entornos cloud, por lo que cada vez más utilizan servicios de esos proveedores, decomisionando infraestructura on-premise (en su propio CPD). Las plataformas Hadoop no son una excepción.

Por este motivo, la principal competencia de Cloudera no es otro fabricante de distribuciones Hadoop, sino los principales proveedores de cloud.

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 desvantajas:

- 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).

Debes conocer

No te preocupes por la diferencia que pueda haber entre las distribuciones Hadoop, la versión de Hadoop "estándar" o los servicios Hadoop en la nube.

En la mayor parte de las actividades, es suficiente conocer Hadoop "estándar" ya que el resto de alternativas apenas difieren y lo aprendido en una sirve perfectamente para otra.

Autoevaluación

Imagina que queremos montar una plataforma Hadoop para traer datos de una base de datos relacional (por ejemplo, una base de datos Oracle), almacenarlos y hacer consultas con un lenguaje similar a SQL para calcular una serie de métricas (medias, máximos, etc.).

¿Cuál de las siguientes combinaciones de componentes de Hadoop crees que servirá para llevar a cabo el caso de uso?

- ○ YARN + Sqoop + Hive
- ○ HDFS + Flume + YARN + Impala
- ○ HDFS + YARN + Sqoop + Hive

Incorrecto: necesitamos un componente que almacene los datos, en este caso, HDFS.

Incorrecto: necesitamos un componente para traer los datos de la base de datos relacional, que sería Sqoop en lugar de Flume.

Correcto

Solución

1. Incorrecto
2. Incorrecto
3. Opción correcta

2.3.- Arquitectura.

Hadoop se basa en un modelo de despliegue distribuido, es decir, se instala sobre un conjunto de servidores que trabajan de forma conjunta para efectuar las tareas.

Pese a que hay un conjunto de servidores trabajando en paralelo y de forma conjunta, para un usuario externo todos ellos actúan como si fuera una sola máquina, es decir, un usuario del sistema de ficheros (HDFS) verá la estructura de directorios, subdirectorios y ficheros, pero no tendrá que conocer en qué servidores está cada fichero, o simplemente, no tendrá constancia de que por debajo se están almacenando en diferentes servidores. Lo mismo ocurre con cualquier otro componente que se ejecuta en toda la infraestructura, ya que para todos ellos, para un usuario externo se usa la funcionalidad ofrecida sin necesidad de tener conocimiento de la infraestructura que da servicio a la funcionalidad.

Debes conocer

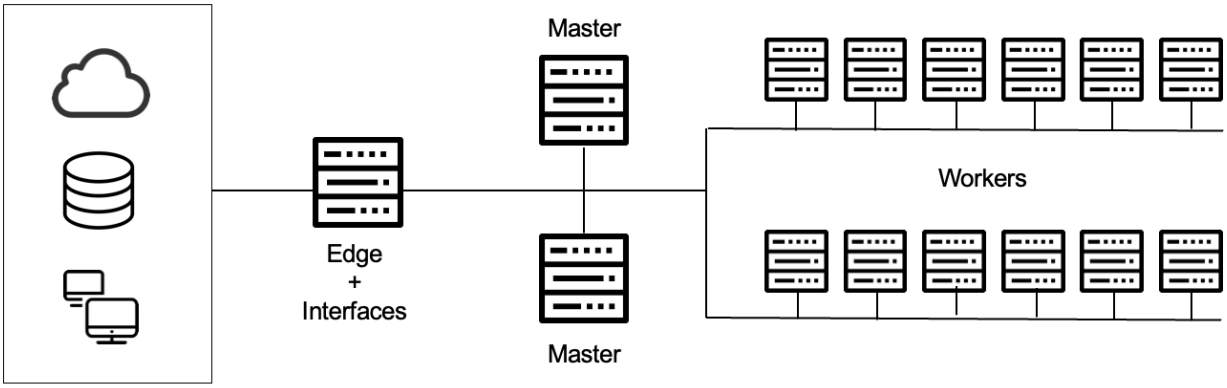
Al un conjunto de servidores que trabajan en conjunto para implementar las funcionalidades de Apache Hadoop se le denomina **clúster**, y a cada uno de los servidores que forman parte del clúster se le denomina **nodo**.

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.

Como hemos comentado, la arquitectura de Hadoop se basa en el trabajo de un conjunto de nodos de forma conjunta. Además, estos 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 continuación se muestra de forma gráfica cómo es la arquitectura de Apache Hadoop:



Íñigo Sanz (Dominio público)

A continuación vamos a ver cómo esta arquitectura permite conseguir algunas de las características que habíamos comentado en el apartado anterior:

Alta disponibilidad o tolerancia a fallos

Esta característica indica que en caso de que se produzca un fallo en alguno de los nodos, el sistema sigue funcionando y no se pierden datos. El fallo puede deberse a la caída de un nodo por completo, por ejemplo, por un fallo de la red o de la fuente de alimentación, o parcial, por ejemplo, por la rotura de un disco.

Hadoop se basa en la suposición de que el hardware falla, y más teniendo en cuenta que un clúster Hadoop puede tener miles de nodos.

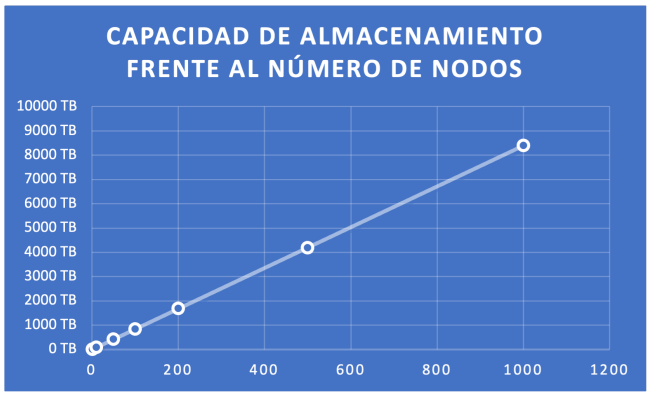
La forma de conseguir la tolerancia a fallos se realiza de diferentes formas:

- En el caso de los nodos master, lo habitual es que existan dos nodos maestros por servicio (HDFS, YARN, etc.), actuando en modo activo-pasivo, es decir, un nodo se ocupa de toda la operativa (activo o primario), mientras que el otro (pasivo o secundario) lo único que realiza es hacer una copia de todo lo que está realizando en activo. En caso de caída del nodo activo, el pasivo toma el control y como dispone de una copia exacta de todos los datos del activo, es capaz de continuar su trabajo hasta que el error en el activo pueda subsanarse.
- Para los nodos worker, la tolerancia a fallos se consigue de forma diferente para el almacenamiento y para el procesamiento:
  - Para conseguir tolerancia a fallos en el almacenamiento, Hadoop hace duplicados de los datos, de manera que cualquier dato dispone de varias réplicas en distintos nodos. En caso de que uno de los nodos se caiga o un disco se corrompa, el resto de réplicas garantizan que se pueda seguir disponiendo de esos datos. En módulos posteriores veremos en detalle cómo funciona la replicación.
  - Para conseguir la tolerancia a fallos en la ejecución de aplicaciones, el nodo maestro es quien ordena a cada nodo worker qué tarea o parte de una tarea debe realizar y va consultando continuamente a todos los nodos worker por su estado. En caso de que alguno se haya caído en mitad de la ejecución de una tarea o subtarea, cogerá este fragmento de trabajo y se lo enviará a otro nodo para que lo realice.
- En el caso de los nodos frontera o edge, al ser servicios externos a Hadoop, no suelen ser críticos, pero en cualquier caso, se suelen montar varios servicios en diferentes nodos para que en caso de caída de un nodo frontera, los usuarios puedan utilizar los servicios en otro nodo.

Escalabilidad

El número de nodos worker se puede incrementar si se desea aumentar la capacidad de almacenamiento o de ejecución de Hadoop. Asimismo, se puede realizar lo opuesto si se desea reducir la capacidad, disminuyendo el número de nodos. Será el nodo maestro el que disponga de un listado de los nodos activos en el clúster, y redistribuirá la carga de trabajo (almacenamiento o ejecución) entre los nodos disponibles.

Añadir nuevos nodos suele ser un proceso sencillo, aunque requiere un rebalanceo de la carga a menudo, para que los nuevos nodos equilibren el trabajo del resto en caso de una ampliación, o para que los nodos que quedan levantados asuman la carga de los nodos que se han eliminado del clúster en caso de una reducción.



Íñigo Sanz (Dominio público)

Hardware no específico (commodity hardware)

A veces el concepto “hardware commodity” suele confundirse con “hardware de andar por casa”, cuando lo que hace referencia es a hardware no específico, que no tiene unos requerimientos en cuanto a disponibilidad o resiliencia exigentes.

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

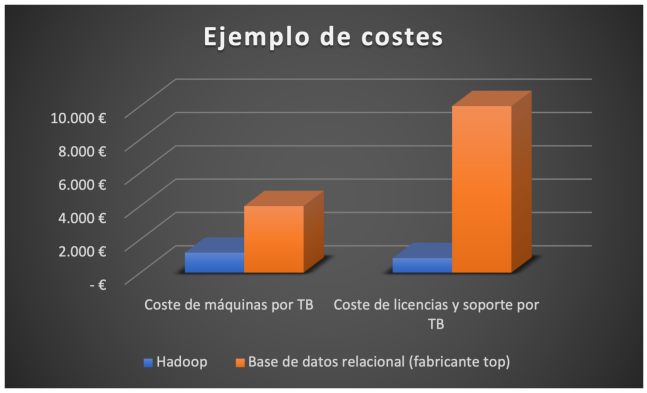
La implantación de una plataforma Hadoop tiene asociado tres tipos de coste:

- 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.

El clúster tendría una capacidad en almacenamiento en torno a 400 terabytes. Para poder compararlo con otro tipo de sistemas de almacenamiento y procesamiento de datos, como por ejemplo, una base de datos relacional tradicional, un sistema tradicional podría tener un coste superior a 1 ó 2 millones de euros anuales sólo en licencias, más el coste del hardware, que sería superior al coste del hardware anteriormente indicado.



Íñigo Sanz (Dominio público)

Autoevaluación

Responde a las siguientes preguntas indicando si las afirmaciones son verdaderas o falsas

Hadoop no dispone de un punto único de fallo

○ Verdadero ○ Falso

Verdadero

Verdadero: todos los servicios de Hadoop y nodos pueden caerse sin que el sistema pueda verse afectado.

Los nodos master deben tener mucha capacidad de almacenamiento para poder guardar una copia de los datos

○ Verdadero ○ Falso

Falso

Falso: los nodos master sólo tienen los datos que permiten saber qué almacena cada nodo o qué tareas hay y cuáles se están ejecutando en cada nodo.

La red no es importante en Hadoop, se puede elegir una red que no sea muy cara

○ Verdadero ○ Falso

Falso

Falso: en cualquier sistema distribuido, la red es un elemento muy importante, por lo que es conveniente invertir en tener una buena red, con un ancho de banda alto, para mejorar el rendimiento de todo el clúster.

2.4.- Beneficios, desventajas y dificultades.

Hadoop es una plataforma de almacenamiento y procesamiento de datos de cualquier tipo y volumen y a un coste razonable cuando el volumen de datos es grande.

No es la única tecnología que da respuesta a casos de uso de Big Data, pero sí la que más casuísticas cubre, es decir, probablemente Hadoop no es la mejor tecnología para gran parte de los casos de uso, pero es la que ofrece un mayor número de posibilidades:

- Para casos de uso de almacenamiento y acceso a los datos en **tiempo real**, soluciones como Apache Cassandra ofrece un rendimiento mejor.
- Para casos de uso de construcción de **cuadros de mando con acceso online**, soluciones como las bases de datos OLAP en memoria, o soluciones mixtas como SAP Hana, ofrecen un mejor desempeño.
- Para casos de uso de **transformación de datos**, las herramientas de ETL son más eficientes.
- Para casos de uso de **machine learning**, soluciones como R tienen una mayor riqueza de operaciones.

Pese a no ser la mejor en los casos de uso anteriores, es suficientemente buena para cubrir la mayor parte de ellos con un buen desempeño, y **todo en la misma plataforma**, en lugar de necesitar una herramienta específica para cada caso de uso. Por este motivo, Hadoop es la plataforma más utilizada en el ámbito Big Data.

OpenClipart-Vectors (Dominio público)

Las compañías adoptan Hadoop porque permite cubrir la mayor parte de casos de uso con un buen desempeño, algo que no ocurre seleccionando otras tecnologías de datos, por ejemplo:

- Seleccionando **MongoDB** se cubren bien los casos de uso operacionales (los que dan soporte a la operativa de la empresa) y algún caso de uso analítico, pero no da cobertura a todos los casos de uso analíticos con buen desempeño, y es difícil trabajar con técnicas de machine learning con datos existentes en MongoDB.
- Seleccionando **SAP HANA** se cubren bien los casos de uso operacionales y analíticos, pero su coste es muy superior a Hadoop y además, no cubre los casos de uso con un volumen de datos muy elevado.
- Seleccionando una **base de datos relacional** se cubren bien los casos de uso operacionales o los casos de uso analíticos, dependiendo de cuál sea la base de datos

elegida, pero la gestión de datos no estructurados es compleja y su capacidad para cubrir casos de uso de alta volumetría (>50 terabytes) está bastante limitada.

Recuerda, **Hadoop se considera una plataforma**, no una herramienta. Es una plataforma porque ofrece la base con la que construir aplicaciones, así como una multitud de herramientas para resolver casos de uso concretos.

Resumen de características de Hadoop

- Almacena y procesa **cualquier tipo de información**, estructurada, no estructurada o semi-estructurada, lo que le da una gran versatilidad.
- Los datos que se incorporan **no necesitan un esquema prefijado**, ya que no obliga a definir en primer lugar cómo son los datos antes de almacenarlos. Esta característica se llama schema-on-read (esquema en lectura), a diferencia de schema-on-write (equema en escritura), que es la característica que tienen las bases de datos relacionales, en las que antes de introducir algún dato, debes detallar cómo es su estructura, con alto nivel de detalle. La característica de Hadoop de esquema en lectura permite almacenar los datos sin tratar, que se conocen como Raw data, y posteriormente procesarlos o analizarlos, lo que aporta mucha rapidez a la hora de implementar proyectos donde los datos cambian a menudo de estructura, o se añaden datos con todo tipo de formatos.
- **Bajo coste** (hardware commodity + código open-source): el coste de una plataforma Hadoop es órdenes de magnitud inferior a otras tecnologías de gestión de datos, como las bases de datos relacionales.
- **Escalabilidad "ilimitada"** y lineal: no significa que con Hadoop no haya límite en cuanto al número de datos a gestionar, ya que el límite lo marca la capacidad que tenga el maestro para poder soportar la carga, pero se dice que el nivel de escalabilidad es ilimitado porque el umbral máximo es muy elevado, en torno a 10.000 nodos en un único clúster, lo que significa unos 50-100 petabytes.
- **Enfoque distribuido**: se trata de un sistema compuesto por muchos nodos, lo que le da un rendimiento excelente para trabajos de mucha complejidad o con un volumen de datos muy elevado, pero un rendimiento inferior cuando el volumen de datos es pequeño o cuando las tareas son sencillas (por ejemplo, consultar un registro aislado).
- **Múltiples herramientas**: como hemos comentado con anterioridad, Hadoop es una plataforma que ofrece una gran capacidad de almacenamiento y múltiples herramienta poder trabajar con los datos.

Problemática asociada

Como cualquier sistema, Hadoop ofrece una serie de ventajas, pero también tiene algunas dificultades o inconvenientes que deben plantearse antes de tomar la decisión sobre si es la herramienta indicada:

- **Requiere nuevos perfiles escasos**: al tratarse de un conjunto de tecnologías con poca antigüedad, no hay una gran cantidad de profesionales con conocimientos en Hadoop o en las herramientas del ecosistema, a diferencia de otras tecnologías de gestión de datos como las bases de datos relacionales. Este punto, aunque puede parecer poco importante, está siendo uno de los factores que más está limitando el uso de Hadoop en las organizaciones. Pese a que cada vez hay más formación disponible y un gran número de personas se está especializando en Hadoop o en el resto de

tecnologías Big Data, la demanda de estos perfiles sigue siendo muy superior a la oferta, lo que está teniendo, entre otras, estas consecuencias:

- Los **costes** se están incrementando, ya que los salarios de los profesionales de Big Data es más elevado que en el resto de tecnologías.
- Existe un riesgo por **rotación** de personal alto, es decir, dada la cantidad de ofertas que reciben los profesionales, las empresas deben adaptarse a una continua fuga de talento.
- Como contrapartida, los **profesionales** de este tipo de tecnologías puede, además de obtener unos salarios y condiciones mejores que en el resto de tecnologías, pueden elegir bien el proyecto o compañía donde desarrollarse profesionalmente, así como su carrera profesional se está viendo acelerada, incrementando su valía en un periodo de tiempo muy corto.

- Hardware commodity no significa reutilizar PCs y opensource no significa que no haya ningún **coste**: aunque el coste de Hadoop es inferior al de otro tipo de tecnologías, a menudo se le presupone unos costes inferiores a la realidad. Es importante hacer una buena estimación económica del coste de adquisición y mantenimiento de estas plataformas, para no tener que paralizar proyectos por falta de presupuesto.
- Hadoop es una **nueva pieza** en las arquitecturas de gestión de datos, y a veces su integración no es sencilla: las empresas deben adaptarse a este nuevo conjunto de tecnologías en diferentes ámbitos:
  - Formación y capacitación: los empleados, que están acostumbrados a manejar otro tipo de tecnologías, deben formarse para poder manejar esta nueva pieza.
  - Hadoop no reemplaza las tecnologías existentes, sino que habitualmente las complementa, por lo que requiere integrarlo en los ecosistemas tecnológicos y las arquitecturas existentes de las empresas. El esfuerzo de integración no es trivial, y requiere diseñar bien el rol y el tipo de integración a realizar.
- El enfoque distribuido tiene sus problemas:
  - **Administración** más compleja (N nodos x M componentes)
  - **Consumo** energético, **espacio** en CPD, …
  - **Menor eficiencia** que enfoques centralizados: como se ha comentado, Hadoop no es en absoluto la mejor tecnología para implementar casos de uso que no tienen una volumetría baja o un procesamiento de datos poco complejo. Es habitual encontrar en el mercado organizaciones que han implementado Hadoop para casos de uso sencillos, y al ponerlos en producción, se dan cuenta de que otro tipo de tecnologías funcionan mejor para el caso de uso. Hadoop debe utilizarse para casos de uso con complejidad "Big Data", bien por el volumen de datos, por la complejidad del procesamiento, por la variedad de los datos o por cualquier otra característica donde las tecnologías tradicionales no puedan cubrir el caso de uso.
  - **Madurez, seguridad y gobierno de datos**: Hadoop es una tecnología con apenas 10 años de uso en las compañías, y no puede compararse en nivel de madurez con las bases de datos relacionales, que surgieron entre los años 70 y 80, especialmente en mecanismos de seguridad, integración o gobierno de datos. Sin ser un problema bloqueante, debe tenerse en cuenta antes de seleccionarlo en casos de uso críticos.

¿Cuándo usar Hadoop?

Oberholster Venita (Dominio público)

Con las características indicadas anteriormente, se puede decir que Hadoop encaja bien en los siguientes casos de uso:

- Cuando el **volumen** de datos es mayor que la capacidad de los sistemas tradicionales (no cabe en una máquina).
- Cuando hay un problema de **variedad** de datos, porque son diversos o porque cambian frecuentemente.
- Cuando se requiere una **escalabilidad** que no pueden ofrecer los sistemas tradicionales, por volumen, por velocidad de proceso, por rendimiento global, y no se requiere un nivel de transaccionalidad elevado.
- Cuando se pretende tener **una plataforma** con la capacidad de almacenamiento y procesamiento de un gran volumen de datos para cubrir diferentes casos de uso (con la misma plataforma).

¿Cuándo no usar Hadoop?

Asimismo, Hadoop no es a priori la mejor tecnología cuando se pretende abordar los siguientes casos de uso:

- Cuando los sistemas tradicionales son capaces de dar soporte a los casos de uso y cuando los formatos/tipos de datos son fijos o no cambian apenas.
- Cuando se tiene requisitos de transaccionalidad muy estrictos, es decir, cuando se pretende cubrir la operativa de una empresa (por ejemplo, en un banco: las transferencias, movimientos, pagos, etc.).
- Cuando sólo se requiere resolver un caso de uso “Big Data” muy específico.

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?

Mostrar 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 preveer 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?

Mostrar 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.

Pregunta Verdadero-Falso

Indica si las siguientes afirmaciones son verdaderas o falsas sobre lo que has aprendido en este apartado.

Hadoop no es probablemente la mejor tecnología para cada caso de uso concreto, pero es bastante buena para la mayoría de casos de uso.

○ Verdadero ○ Falso

Verdadero

Verdadero.

Hadoop es bastante eficiente incluso con pocos datos.

○ Verdadero ○ Falso

Falso

Falso: para escenarios con pocos datos, al ser una tecnología distribuida, es poco eficiente.

Si no sé qué tecnología Big Data implantar para mi empresa porque tengo necesidad de resolver multitud de casos de uso, Hadoop puede ser una buena opción

○ Verdadero ○ Falso

Verdadero

Verdadero: aunque es precipitado prescribir Hadoop sin conocer más detalles, es más difícil equivocarse si hay un número de casos de uso que cubrir elevado, con casos de uso variopintos.

3.- Instalación de Apache Hadoop con Docker

En esta sección vamos a aprender a instalar Apache Hadoop y algunas de las herramientas de su ecosistema en tu equipo. Para ello utilizaremos contenedores de Docker. No te preocupes si no conoces Docker ya que también se explica cómo instalarlo y una guía básica de uso. En la parte final de la guía se explicará cómo usar Python en Jupyter. Esto facilitará crear programas MapReduce en Hadoop en las próximas unidades. Antes de empezar con esta guía debes haber leído los apartados anteriores de esta unidad.

El archivo comprimido que se adjunta contiene tanto en formato "ipynb" como en formato "html" el proceso que hay que seguir para instalar Hadoop en tu equipo. Descomprímelo y abre el fichero "html" en un navegador o el de "ipynb" en un servidor de Jupyter. También puedes reproducir el vídeo en el que se desarrollan paso a paso los contenidos de la guía. Por último, el fichero comprimido se incluye un fichero con enlaces de interés que puedes importar en los marcadores de tu navegador si así lo deseas. Si tienes dudas o te surgen problemas durante el proceso, no dudes en escribir en el foro explicando en detalle qué es lo que ha sucedido.

https://www.youtube.com/embed/KYfpVuQ1VKI

- guia_BDA01.zip (Ventana nueva)