### **HADOOP**

### **Breve historia**

Como podríamos imaginarnos los primeros en encontrarse con problemas de procesamiento, almacenamiento y alta disponibilidad de grandes bancos de información fueron los buscadores y las redes sociales. Con la implementación de sus algoritmos de búsquedas y con la indexación de los datos en poco tiempo se dieron cuenta de que debían hacer algo y ya.

Fue así como nació el sistema de archivos de Google (GFS), un sistema de archivos distribuido capaz de ser montado en clústeres de cientos o incluso de miles de máquinas compuestas por hardware de bajo costo. Entre algunas de sus premisas contaba con que es más óptimo el tratamiento de pocos ficheros de gran tamaño que el tratamiento de muchos pequeños y que la mayor parte de las modificaciones sobre los ficheros suelen consistir en añadir datos nuevos al final en vez de reemplazar los existentes.

De la mano del GFS crearon Map-Reduce del cual hablaremos más adelante pero podríamos decir que es un modelo de programación cuya finalidad es paralelizar o distribuir el procesamiento sobre los datos a través de los clústeres.

Cabe mencionar que Google una vez creo semejante solución público la documentación donde describe aspectos de arquitectura y funcionamiento tanto del GFS como del Map-Reduce el cual sirvió como base para la creación de Hadoop.

Aquí están los links de la documentación publicada por Google:

* The Google File System, Octubre 2003. Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. http://research.google.com/archive/gfs.html
* MapReduce: Simplified Data Processing on Large Clusters, Diciembre 2004. Jeffrey Dean and Sanjay Ghemawat. http://research.google.com/archive/mapreduce.html

Así pues de la mano del gran avance de Google fue como nació Hadoop con Doug Cutting un trabajador de Yahoo!. Gracias a que había sido el creador de distintos proyectos en Open Source con anterioridad distribuyó de la misma manera a Hadoop.

https://wiki.apache.org/hadoop

De igual manera se hace mención a Facebook ya que es parte de la popularización y contribución al desarrollo de Hadoop, pues bien fue uno de los primeros en implementar uno de los clústeres más grandes y se atribuye el desarrollo de <Hive> el cual permite realizar consultas similares al SQL sobre los datos en un ambiente distribuido.

### **¿Qué es Hadoop?**

Es un framework Open Source de Apache que nos permite implementar plataformas Big Data altamente distribuidas, funcionales y escalables sin depender de la inversión de licencias y ni de hardware. Gracias a que está escrito en el lenguaje de programación Java disponemos de diversas aplicaciones o proyectos adicionales que le potencian además de su propia adaptación del modelo de programación Map-Reduce para el procesamiento de grandes bancos de datos incluyendo estructurados y no estructurados.

Hadoop está diseñado para crear aplicaciones que procesen grandes volúmenes de datos de manera distribuida a través de Map-Reduce. Además gracias a que trabaja con almacenamiento y procesamiento local (pero distribuido) nos permite trabajar tanto con Clústeres de un solo nodo como por miles de nodos ofreciendo un alto nivel de tolerancia a errores.

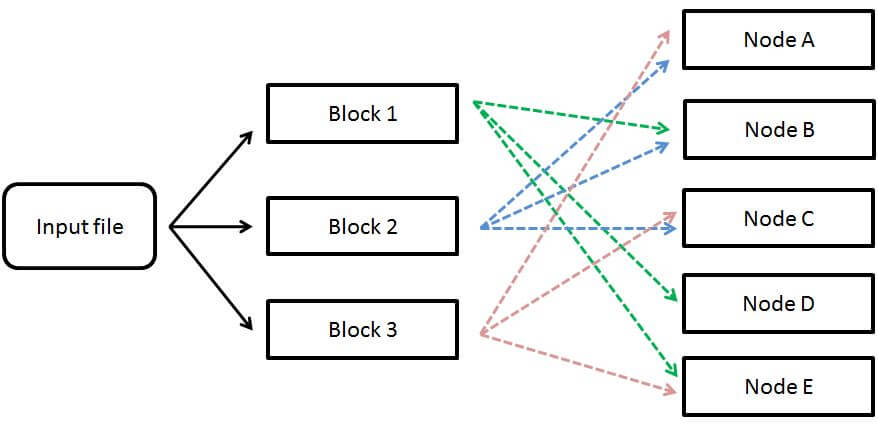
### **Componentes de Hadoop**

Consta principalmente de dos componentes los cuales se dividen en almacenamiento/distribución de los datos y el procesamiento de los mismos:

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

Es un sistema de archivos distribuidos el cual permite difundir los datos a través de cientos o miles de nodos para su procesamiento. Aquí es donde se proporciona redundancia (Los datos están repetidos o replicados en varios nodos) y tolerancia a fallos (Si falla algún nodo se reemplaza automáticamente).

En su funcionamiento el sistema de archivos HDFS divide los datos en bloques donde a su vez cada bloques se replica en distintos nodos de manera que la caída de un nodo no implique la pérdida de los datos que éste contiene. De esta manera se facilita el uso de modelos de programación como Map-Reduce, ya que se puede acceder a varios bloques de un mismo fichero en forma paralela.



### **MapReduce**

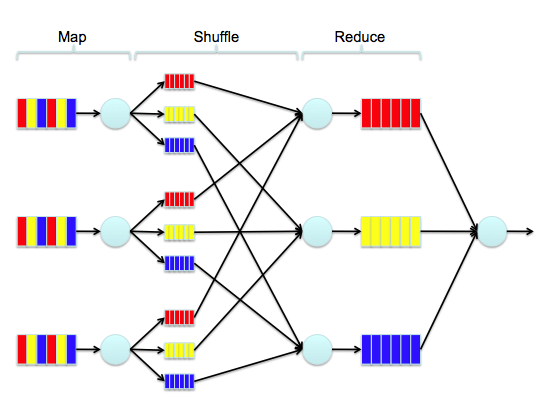
Es el corazón de Hadoop el cual permite el fácil desarrollo de aplicaciones y algoritmos bajo el lenguaje Java para el procesamiento distribuido de grandes cantidades de datos.

Dentro del ecosistema las aplicaciones desarrolladas para el framewrok Map-Reduce se conocen como Jobs, éstos se componen de las siguientes funciones:

**Map (Mapeo):** Encargada de la división de las unidades de procesamiento a ejecutar en cada nodo y de su distribución para su ejecución en paralelo. Aquí a cada llamada se le asignará una lista de pares key/value.

**Shuffle and sort (Combinación y Orden):** Aquí se mezclan los resultados de la etapa anterior con todas las parejas clave/valor para combinarlos en una lista y a su vez se ordenan por clave.

**Reduce (Reducir):** Aquí se reciben todas las claves y listas de valores haciendo la agregación de las mismas si fuese necesario.



### **Tipos de instalación**

Antes de continuar es necesario mencionar que Hadoop fue diseñado para ejecutar procesamiento distribuido por lo cual tendremos nodos maestros (Para cubrir las funciones de control del almacenamiento y procesamiento) y nodos esclavos (donde se almacenan o tratan los datos).

Una de las grandes ventajas de Hadoop es que inicialmente no necesitamos disponer de grandes requerimientos de hardware para montar nuestra propia aplicación, gracias a que su escalabilidad nos permite crecer conforme lo requieran las exigencias de nuestro proyecto.

Es importante recalcar que no es lo mismo realizar el conteo de 1000 palabras a contar todas las palabras de todos libros de una biblioteca nacional, debemos ser realistas, así que si te encuentras con una solución real de Big Data donde debes procesar Petabytes de datos si será necesario contar con exigentes requerimientos.

### **Stand Alone (un solo nodo)**

La instalación Stand Alone nos permite configurar un despliegue de Hadoop donde todos los servicios tanto de maestro como de esclavo se ejecutan en un solo nodo al mismo tiempo que solo trabaja con un thread.

Una de las grandes ventajas es que nos puede servir para probar aplicaciones sin tener que preocuparnos de concurrencias de ejecución, además es una buena manera de entrar en el mundo de Hadoop pues experimentaras aspectos a tener en cuenta de instalación y configuración los cuales seguramente necesitaras en un futuro.

### **Pseudo-distribuida**

Esta instalación al igual que la anterior también se monta sobre un solo nodo, todos sus servicios corren sobre éste pero la diferencia es que permite ejecutar múltiples threads.

Gracias a que ejecutamos aplicaciones multi-threads podemos experimentar la ejecución de aplicaciones sin necesidad de disponer de múltiples computadoras  pues aquí empezamos a aprovechar mucho mejor los distintos núcleos de nuestro procesador.

### **Totalmente Distribuida**

Tal como su nombre lo indica es aquí donde instalamos y configuramos un ambiente completamente distribuido donde dispondremos de un maestro y varios esclavos aprovechando completamente la paralización.

Este es el tipo de instalación para un ambiente de producción donde consideraremos grandes racks y nodos bien dotados de procesador y memoria para divisar los verdaderos beneficios de Hadoop.

**Ecosistema Hadoop**

Gracias a la creciente comunidad Open Source existen distintos proyectos y herramientas que ofrecen funcionalidades adicionales las cuales son consideradas parte de un gran ecosistema pensado en apoyar las distintas etapas de un proyecto Big Data.

A continuación realizo una clasificación personal de las mismas a modo de orden y guía para entenderlas mucho más fácil.

Cada una se acompaña del link oficial y si está disponible un link donde puedes aprender más detalle sobre la misma.

### **Chukwa**

Es un proyecto construido para capturar y analizar grandes volúmenes de datos principalmente logs. Debido a que está construido sobre Hadoop hereda escalabilidad y robustez con el uso de HDFS y Map-Reduce al mismo tiempo que provee adicionalmente un grupo de herramientas flexibles y potentes para visualizar, controlar y analizar los datos.

http://chukwa.apache.org/



### **Flume**

Es una herramienta distribuida para la recolección, agregación y transmisión de grandes volúmenes de datos de diferentes orígenes altamente configurable. Ofrece una arquitectura basada en la transmisión de datos por streaming altamente flexible y configurable pero a la vez simple de manera que se adapta a distintas situaciones tales como monitorización logs (control de calidad y mejora de la producción), obtención de datos desde las [redes sociales](https://www.youtube.com/watch?v=PdY31i25SL0) (Sentiment Analysis y medición de reputación) o mensajes de correo electrónico.

<https://flume.apache.org/>



### **Sqoop**

Es una de las herramientas que deberías tener en cuenta si deseas potenciar tu sistema BI pues su funcionalidad permite mover grandes cantidades de datos entre Hadoop y bases de datos relacionales al mismo tiempo que ofrece integración con otros sistemas basados en Hadoop tales como Hive, HBase y Oozie <links>.

Utilizando el framework Map-Reduce transfiere los datos del DW en paralelo hacia los distintos Clústeres de manera que una vez ahí puede realizar análisis más potentes que el análisis tradicional.

<http://sqoop.apache.org/>

sqoop-logo

### **Hive**

Es una herramienta data warehousing que facilita la creación, consulta y administración de grandes volúmenes de datos almacenados en Hadoop. Cuenta con su propio lenguaje derivado del SQL, conocido como Hive QL, el cual permite realizar las consultar sobre los datos utilizando Map-Reduce para poder paralelizar las tareas. Por esta misma razón, se dice que Hive lleva las bases de datos relacionales a Hadoop.

Otra gran ventaja es nuestro camino a la evolución del BI es que posee drivers de conexión tales como JDBC/ODBC por lo que facilitaría notablemente la integración con nuestros sistemas proporcionándonos extensión en análisis y procesamiento sin cargar el proceso diario.

<http://hive.apache.org/>



### **Hbase**

Es la base de datos de Hadoop distribuida y escalable. Su principal uso se encuentra cuando se requieren escrituras/lecturas en tiempo real y acceso aleatorio para grandes conjuntos de datos. Debido a que su base es hadoop adquiere las sus capacidades y funciona sobre HDFS. Puedes almacenar en un amiente distribuido tablas sumamente grandes incluso hablando de billones de registros por millones de columnas, la manera de soportar esta cantidad de datos es debido a que es una base NoSQL de tipo Columnar por lo cual no es posible realizar consultas SQL.

<http://hbase.apache.org/>



### **Mahout**

Este proyecto nos permite desarrollar algoritmos escalables de Machine Learning y Data Mining sobre Hadoop. Soporta algoritmos como recomendación, clustering, clasificación y filtro colaborativo, también si es el caso podremos crear algoritmos para encontrar patrones, que aprendan sobre  los datos y que los  clasifique una vez termine su fase de aprendizaje.

http://mahout.apache.org/



### **Pig**

Este proyecto nos permite analizar grandes volúmenes de datos mediante el uso de su propio  leguaje de alto nivel llamado **PigLatin.**Sus inicios fueron en Yahoo donde sus desarrolladores pensaban que el Map-Reduce era de muy bajo nivel y muy rígido por lo cual podías tardar mucho tiempo en la elaboración y manutención.

Así pues nace Pig  con su propio lenguaje y trabaja sobre Hadoop traduciendo las consultas del usuario a Map-Reduce sin que éste siquiera lo note. De esta manera provee un entorno fácil de programación convirtiendo las paralelizaciones en dataflows, un concepto mucho más sencillo para el usuario del negocio.

Pig tiene dos componentes: su lenguaje PigLatin y su entorno de ejecución.

Pig provee un enfoque más analítico que a la construcción.

Debido a su fácil y potente uso es usado en procesos de ETL y en la manipulación y análisis de datos crudos.

<http://pig.apache.org/>



### **Oozie**

Proyecto el cual nos permite planificar workflows para soluciones que realizan procesos o tareas Hadoop. Al igual que Pig, está orientado al usuario no experto por lo cual le permite definir fácilmente flujos de trabajo complejos sobre los datos.

Oozie funciona como un motor de workflows a modo de servicio que permite lanzar, parar, suspender, retomar y volver a ejecutar una serie de trabajos Hadoop  (tales como Java Map-Reduce, Streaming Map-Reduce, Pig, Hive, Sqooq…) basándose en ciertos criterios, como temporales o de disponibilidad de datos. Los flujos de trabajo Oozie son grafos no cíclicos directos -también conocidos como DAGs- donde cada nodo es un trabajo o acción con control de dependencia, es decir, que una acción no puede ejecutarse a menos que la anterior haya terminado.

<http://oozie.apache.org/>

oozie_logo

### **Zooquiper**

Es un proyecto de Apache el cual brinda una infraestructura centralizada y servicios que permiten la sincronización del clúster. Zooquiper en pocas palabras se encarga de administrar y gestionar la coordinación entre los distintos procesos de los sistemas distribuidos.

https://zookeeper.apache.org/



### **Avro**

Avro, es un sistema de serialización  de datos creado por Doug Cutting, el padre de Hadoop.

Debido a que podemos encontrar distintos formatos de datos dentro de Hadoop, Avro se ocupa de que dichos formatos puedan ser procesados por distintos legajes de programación por ejemplo Java, C, C++, Python, Ruby y C#. El formato que utiliza para serializar el JSON gracias a su portabilidad y fácil lectura.

https://avro.apache.org/



### **Hue**

Hue es una herramienta enfocada en los administradores de las distribuciones Hadoop proporcionando una interfaz web para poder trabajar y administrar las distintas herramientas instaladas. Desde aquí puedes cargar o visualizar datos, programar y ejecutar consultas Pig o SQL, realizar búsquedas e incluso programar en pocos pasos un flujo de datos.

Una funcionalidad que independientemente de las herramientas que tengas instaladas en tu proyecto Big Data con Hadoop no puede faltar.

<http://gethue.com/>



Fuente: <http://blog.jacagudelo.com/hadoop-introduccion-componentes-y-ecosistema/>