**1 Que es el Big Data**

 Estamos viviendo la época de la revolución del **Big Data**, donde los grandes volúmenes de datos, usados para trabajar, han superado con creces, la capacidad de procesamiento de un simple host. El **Big Data** nace para solucionar estos problemas:

* Como almacenar y trabajar con grandes volúmenes de datos.
* Y la más importante, como poder interpretar y analizar estos datos, de naturaleza muy dispar.

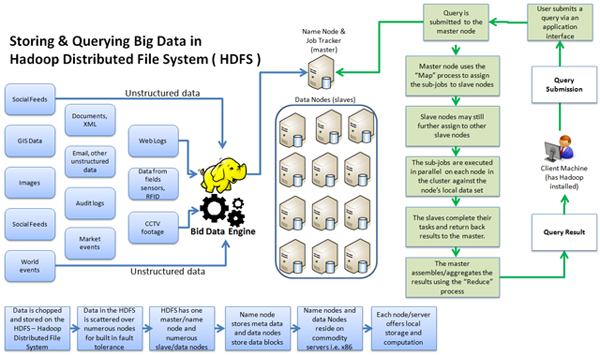
Si miramos alrededor nuestro, vemos que cualquier dispositivo que usamos genera datos, estos pueden ser analizados actualmente. De esta gran cantidad de datos que tenemos a nuestro alcance, sólo el 20% se trata de información estructura y el 80% son datos no estructurados. Estos últimos añaden complejidad en la forma que se tienen que almacenar y analizar.

**Hadoop** aparece en el mercado como una solución para estos problemas, dando una forma de almacenar y procesar estos datos.

**2 Hadoop y su arquitectura**

**2.1 Qué es Hadoop?**

**Apache Hadoop** es un framework que permite el procesamiento de grandes volúmenes de datos a través de clusters, usando un modelo simple de programación. Además su diseño permite pasar de pocos nodos a miles de nodos de forma ágil. Hadoop es **un sistema distribuido** usando una arquitectura **Master-Slave**, usando para almacenar su **Hadoop Distributed File System** (HDFS) y algoritmos de **MapReduce** para hacer cálculos.



**2.2 Arquitectura principal de Hadoop**

**2.2.1 HDFS**

**HDFS** es el sistema de almacenamiento, es un sistema de ficheros distribuido. Fue creado a partir del **Google File System** (GFS). HDFS se encuentra optimizado para grandes flujos y trabajar con ficheros grandes en sus lecturas y escrituras. Su diseño reduce la E/S en la red. La escalabilidad y disponibilidad son otras de sus claves, gracias a la replicación de los datos y tolerancia a los fallos. Los elementos importantes del cluster:

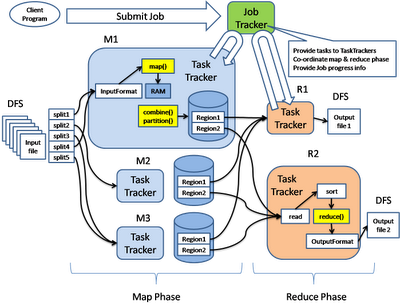
* **NameNode**: Sólo hay uno en el cluster. Regula el acceso a los ficheros por parte de los clientes. Mantiene en memoria la metadata del sistema de ficheros y control de los bloques de fichero que tiene cada DataNode.
* **DataNode**: Son los responsables de leer y escribir las peticiones de los clientes. Los ficheros están formados por bloques, estos se encuentran replicados en diferentes nodos.

**2.2.2 MapReduce**

**MapReduce** es un proceso batch, creado para el proceso distribuido de los datos. Permite de una forma simple, paralelizar  trabajo sobre los grandes volúmenes de datos, como combinar web logs con los datos relacionales de una base de datos OLTP, de esta forma ver como los usuarios interactúan con el website.

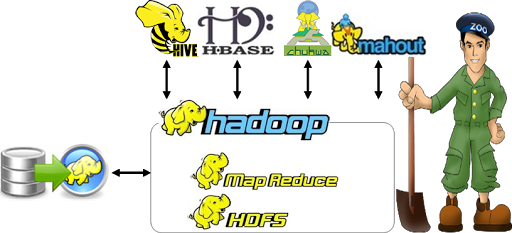
El modelo de MapReduce simplifica el procesamiento en paralelo, abstrayéndonos de la complejidad que hay en los sistemas distribuidos. Básicamente las funciones **Map** transforman un conjunto de datos a un número de pares **key/value**. Cada uno de estos elementos se encontrará ordenado por su clave, y la función reduce es usada para combinar los valores (con la misma clave) en un mismo resultado.

Un programa en MapReduce, se suele conocer como Job, la ejecución de un Job empieza cuando el cliente manda la configuración de Job al **JobTracker**, esta configuración especifica las funciones **Map**, **Combine** (shuttle) y **Reduce**, además de la entrada y salida de los datos.



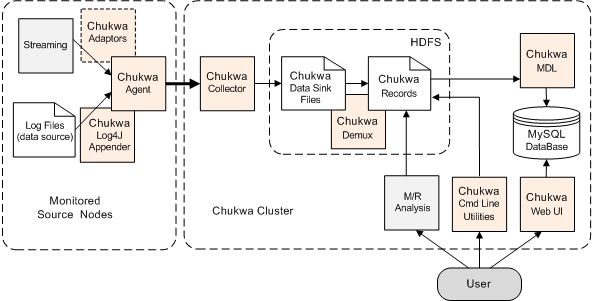
**3 El escosistema de Hadoop**

En Hadoop tenemos un ecosistema muy diverso, que crece día tras día, por lo que es difícil saber de todos los proyectos que interactúan con Hadoop de alguna forma. A continuación sólo mostraremos los más comunes.



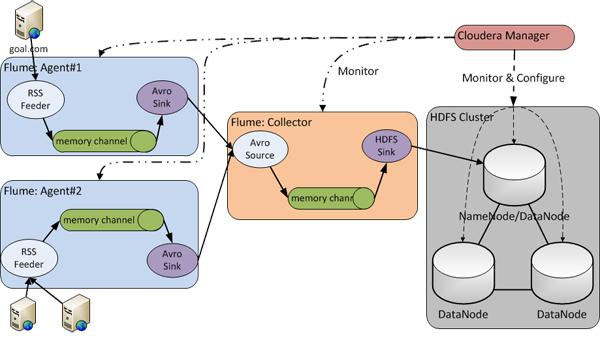
**3.1 Chukwa (**[**http://incubator.apache.org/chukwa/**](http://incubator.apache.org/chukwa/)**)**

**Chukwa** es un sistema de captura de datos y framework de análisis que trabaja con Hadoop para procesar y analizar grandes volúmenes de logs. Incluye herramientas para mostrar y monitorizar los datos capturados.



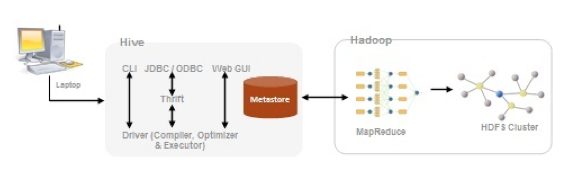
**3.2 Apache Flume (**[**http://flume.apache.org/**](http://flume.apache.org/)**)**

**Apache Flume** es un sistema distribuido para capturar de forma eficiente, agregar y mover grandes cuantidades de datos log de diferentes orígenes (diferentes servidores) a un repositorio central, simplificando el proceso de recolectar estos datos para almacenarlos en Hadoop y poder analizarlos. **Flume** y **Chukwa** son proyectos parecidos, la principal diferencia es que **Chukwa** está pensado para ser usado en **Batch**.



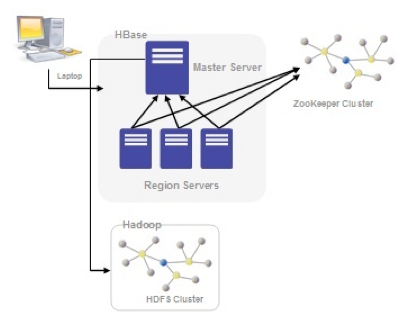
**3.3 Hive (**[**http://hive.apache.org/**](http://hive.apache.org/)**)**

**Hive** es un sistema de Data Warehouse para Hadoop que facilita el uso de la agregación de los datos, ad-hoc queries, y el análisis de grandes datasets almacenados en Hadoop. **Hive** proporciona  métodos de consulta de los datos usando un lenguaje parecido al SQL, llamado **HiveQL**. Además permite de usar los tradicionales Map/Reduce cuando el rendimiento no es el correcto. Tiene interfaces **JDBC/ODBC**, por lo que empieza a funcionar su integración con herramientas de BI.



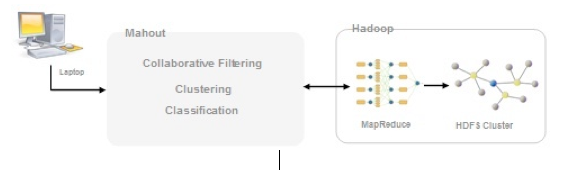
**3.4 Apache HBase (**[**http://hbase.apache.org/**](http://hbase.apache.org/)**)**

**HBase**, se trata de la base de datos de Hadoop. **HBase** es el componente de Hadoop a usar, cuando se requiere escrituras/lecturas en tiempo real y acceso aleatorio para grandes conjuntos de datos. Es una base de datos orientada a la columna, eso quiere decir que no sigue el esquema relacional. No admite SQL.



**3.5  Apache Mahout (**[**http://mahout.apache.org/**](http://mahout.apache.org/)**)**

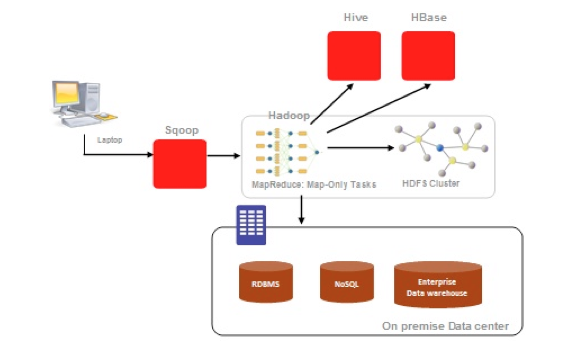
**Apache Mahout**, es un proyecto para crear aprendizaje automático y data mining usando Hadoop. Es decir, Mahout nos puede ayudar a descubrir patrones en grandes datasets. Tiene algoritmos de **recomendación**, **clustering** y **clasificación**.



**3.6  Apache Sqoop (**[**http://sqoop.apache.org/**](http://sqoop.apache.org/)**)**

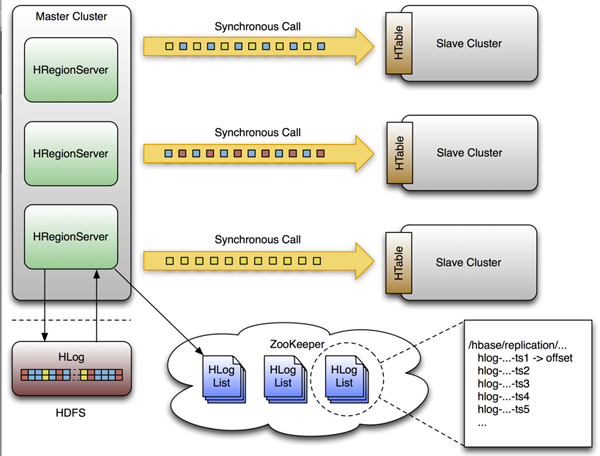
**Apache Sqoop** (“Sql-to-Hadoop”), es una herramienta diseñada para transferir de forma eficiente bulk data entre Hadoop y sistemas de almacenamiento  con datos estructurados, como bases de datos relacionales. Algunas de sus características son:

* Permite importar tablas individuales o bases de datos enteras a HDFS.
* Genera clases Java que permiten interactuar con los datos importados.
* Además, permite importar de las bases de datos SQL a Hive.



**3.7 Apache ZooKeeper (http://zookeeper.apache.org/)**

**Zookeeper** es un proyecto de Apache que proporciona una infraestructura centralizada y de servicios que permiten la sincronización del cluster. ZooKeeper mantiene objetos comunes que se necesiten en grandes entornos de cluster. Algunos ejemplos de estos objetos son información de la configuración, jerarquía de nombres…



**3.8  Apache Lucene (**[**http://lucene.apache.org/core/**](http://lucene.apache.org/core/)**)**

**Lucene**, se trata de una librería escrita en Java, para buscar textos. **Lucene** permite indexar cualquier texto que deseemos, permitiéndonos después encontrarlos basados en cualquier criterio de búsqueda. Aunque Lucene sólo funciona en texto plano, hay plugins que permite la indexación y búsqueda de contenido en documentos Word, Pdf, XML o páginas HTML.

**3.9  Apache Pig (**[**http://pig.apache.org/**](http://pig.apache.org/)**)**

**Apache Pig**, inicialmente desarrollado por Yahoo, permite a los usuarios de Hadoop centrarse más en el análisis de los datos y menos en la creación de programas MapReduce. Para simplificar el análisis proporciona un lenguaje procedural de alto nivel. Su nombre viene de la siguiente analogía, al igual que los cerdos comen de todo, el lenguaje de programación Pig está pensado para poder trabajar en cualquier tipo de datos. Pig consta de dos componentes:

* El lenguaje en si, llamado **PigLatin**.
* El **entorno de ejecución**, donde los programas PigLatin se ejecutan.

**3.10  Jaql (**[**https://code.google.com/p/jaql/**](https://code.google.com/p/jaql/)**)**

**Jaql** es un lenguaje de consulta funcional y declarativo que facilita la explotación de información organizada en formato **JSON** (JavaScript Object Notation), e incluso en archivos semi-estructurados de texto plano. Diseñado inicialmente por IBM. Jaql permite hacer select, join, group y filtrar datos almacenados en HDFS. El objetivo de Jaql es que el desarrollador de aplicaciones de Hadoop pueda concentrarse en qué quiere obtener, y no en cómo lo tenga que obtener. Jaql analiza la lógica y la distribuye en Map y Reduce según sea necesario.

**3.11  Apache Avro (**[**http://avro.apache.org/**](http://avro.apache.org/)**)**

**Avro**, es un sistema de serialización de datos. En los proyectos en Hadoop, suele haber grandes cuantidades de datos, la serialización se usa para procesarlos y almacenar estos datos, de forma que el rendimiento en tiempo sea efectivo. Esta serialización puede ser en texto en plano, JSON, en formato binario. Con Avro podemos almacenar y leer los datos fácilmente desde diferentes lenguajes de programación. Está optimizado para minimizar el espacio en disco necesario para nuestros datos.

**3.12  Apache UIMA (**[**http://uima.apache.org/**](http://uima.apache.org/)**)**

**UIMA** (Unstructured Information Management Applications) es un framework para analizar grandes volúmenes de datos no estructurados, como texto, video, datos de audio, etc… y obtener conocimiento que sea relevante para el usuario final. Por ejemplo a partir de un fichero plano, poder descubrir que entidades son personas, lugares, organizaciones, etc…

