Hadoop

Lesly Reyes

¿Por qué crear Hadoop ?

Contexto

- Big Data.
- Ancestry.com 2.5 petabytes
- The Large Hadron Collider (LHC) 15 petabytes por año.
- Facebook maneja 10 billones de fotos.

Consecuencias

- En un disco duro con velocidad de 100MB/S de 1 tera, 3 horas.
- Ya no es una opción aumentar nuestro tiempo de procesamiento.
  - El bandwidth nos limita.
  - Límites físicos de hardware.

Solución

- Paralelizar el procesamiento de datos.
  - Múltiples procesadores.
  - Múltiples computadores.
- No mover la data.

Desarrolladores

Michael
Cafarella

Doug
Cutting

Principios de diseño de Hadoop

- Un solo core modular y extensible.

- El cómputo se debe mover a la data.
  - Para reducir la latencia y bandwidth

- Performance debe escalar linealmente.
  - Cambios proporcionales en la capacidad del sistema cuando cambian los recursos.

- El sistema se debe gestionar y auto-reparar de forma autónoma.
  - Automático y transparente al momento de la falla.

Hadoop

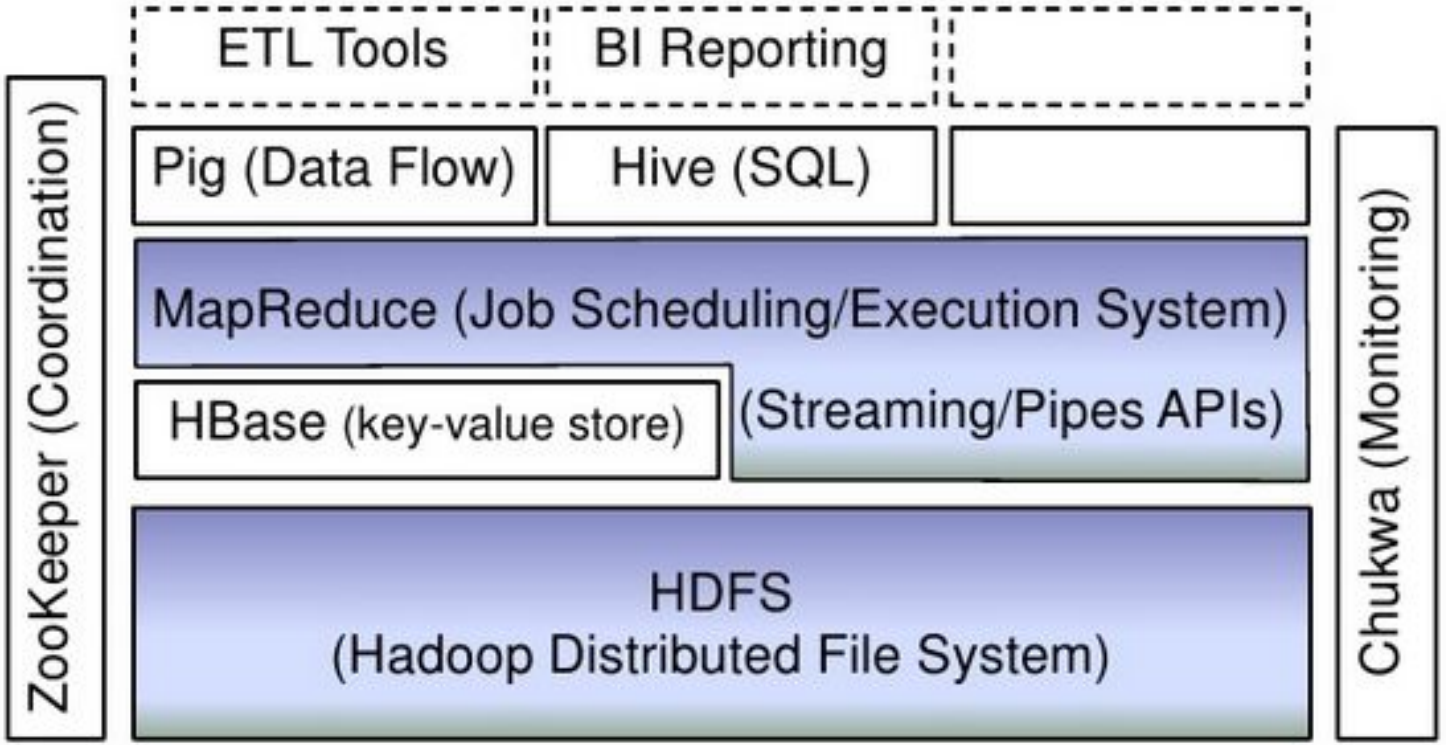
Un framework escalable y tolerante a fallas para guardar y procesar una gran cantidad de datos:

- Arquitectura Master (name-node) - Slave
- Open Source, licencia Apache.
- Funciona con datos estructurados y no-estructurados.

Módulos

- **Hadoop Commons**: Librerías y utilidades para usar hadoop.
- **Hadoop Distributed file system(HDFS)**: Sistema de archivos distribuido.
- **Hadoop YARN**: Sistema para manejar los recursos y trabajos en un cluster.
- **Hadoop MapReduce**: Una implementación de MapReduce para big data.

Arquitectura



Hadoop Distributed File System

- HDFS guarda la metadata del sistema y la data de la aplicación **separadamente**.
- La metadata se guarda en un servidor dedicado llamado **NameNode**. La data de la aplicación se guarda en otros servidores llamados **DataNodes**.
- Estos servidores están todos conectados entre ellos y se comunican a través de un protocolo basado en **TCP.**

Hadoop Distributed File System

- El HDFS client soporta operaciones de leer, escribir y borrar archivos/directorios.
- Single-writer, multiple-readers.
- Este debe mandar heartbeats constantemente para renovar el lease(cliente que escribe).

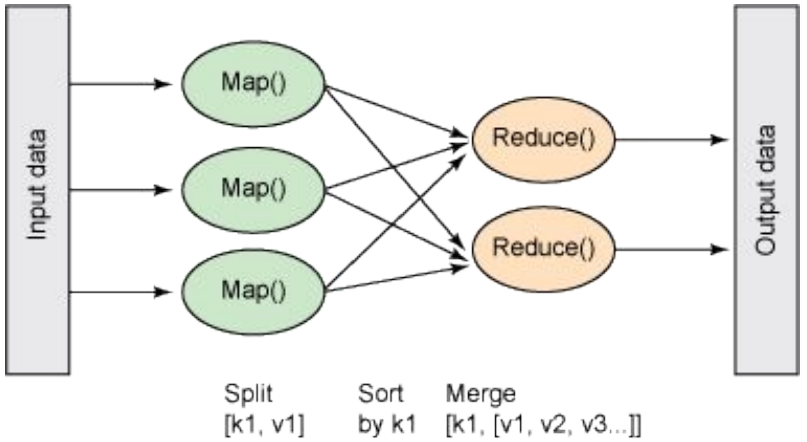
MapReduce

- Patentado por Google Framework.
- Procesamiento distribuido de grandes datos.

```
map (in_key, in_value) ->
  list(out_key, intermediate_value)
reduce
  (out_key, list(intermediate_value)) -
  > list(out_value)
```

MapReduce implementación

- **Job tracker**
  - Separa en map y reduce las tareas.
  - Administra las tareas en un nodo del cluster.
- **Task tracker**
  - Corre las tareas periódicamente de map y reduce. Envía reportes al jobtracker.



HBase

- Base de datos NoSQL distribuido de Hadoop
- Complemento al uso de HDFS
- Versión Open Source de Google BigTable
- Orientado a Columnas
- Key/Value

PIG

- PIG es un Lenguaje de programación para crear programas de análisis sobre datos.
- Se compila en batches de trabajos MapReduce.
- Trabaja sobre HBase.
- Parecido a SQL.

```
grunt> A = LOAD 'student' USING
PigStorage() AS (name:chararray, age:int,
gpa:float);
grunt> B = FOREACH A GENERATE name;
```

HIVE

- Manejo de data estructurada.
- Infraestructura de almacenamiento de datos.
- Syntax like SQL.
- Se ejecuta con mapreduce.
- Agrupación, consulta y análisis de datos.

```
hive> SELECT a.foo FROM invites a WHERE
a.ds='<DATE>';
```

ZOOKEEPER

- Servicio para la coordinación de procesos distribuidos.
- Confiable.
- Brinda soluciones para grandes sistemas distribuido.
- Consenso distribuido: gestión de grupos, protocolos de presencia y elección de líder.
- Alta disponibilidad con servicios redundantes.

Beneficios de usar hadoop

- Reduce los costos de guardar y procesar grandes cantidades de datos.
- Provee una solución escalable y económica.
- Es una solución confiable al momento de guardar los datos.
- Puede procesar gran cantidad de datos de forma paralela en servidores de standard industrial.

Conclusión

Hadoop provee una solución robusta a la tolerancia a fallas, lo que provee al consumidor de un servicio confiable. No obstante presenta un costo de implementación no despreciable.

El beneficio de implementar map reduce de forma distribuida otorga los beneficios de programación paralela, el cómputo va a la data, permite ser ejecuta en gran cantidad de datos.

Referencias

- Chuwka:
  - Link: https://wiki.apache.org/hadoop/Chukwa
- Tutorial:
  - Link:http://enos.itcollege.ee/~jpoial/allalaadimised/reading/Apache-Hadoop-Tutorial.pdf
- Para HDFS
  - Link:https://ieeexplore.ieee.org/abstract/document/5496972/