

# GPUL

## XIII Xornadas Libres



## Creación de un clúster de Hadoop con Cloudera

David Albela Pérez (@elmadno)



# Licencia



Creación de un clúster de Hadoop con Cloudera by David Albela Pérez is licensed under a Creative Commons Reconocimiento-CompartirIgual 4.0 Internacional License.



# Índice

- **Introducción**
- Arquitectura
- Requisitos
- Instalación
- HDFS
- MapReduce
- Ecosistema
- Alta Disponibilidad
- Resumen



# Computación distribuida

- Procesamiento paralelo en varios cientos o miles de nodos
  - Frameworks como [Open MPI](#) o PVM
- Enfocados a distribuir la carga de procesamiento
  - Nodos con alto poder computacional
  - Sistemas separados del almacenamiento

## Problemas

- Cuello de botella con gran cantidad de datos
- Hardware caro y programas complejos con sincronización



# Grace Murray Hopper

- “In pioneer days they used oxen for heavy pulling, and when one ox couldn't budge a log, they didn't try to grow a larger ox”
- “We shouldn't be trying for bigger computers, but for more systems of computers.”



# Hadoop

- Inspirado en los papers de GFS y MapReduce
- Escalabilidad horizontal
  - Comunicación entre nodos mínima
  - Añadir más nodos incrementa la capacidad y rendimiento
- Construido para *commodity hardware*



# Hadoop

- Plataforma Open Source Java para soluciones Big Data
- Doug Cutting (Lucene, Nutch)
- Componentes
  - Commodity Hardware
  - Sistema de ficheros distribuido (HDFS)
  - Paradigma MapReduce
- Amplio ecosistema (Sqoop, Hive, Pig, Hbase, Mahout, etc.)



# Historia

- 2003, [Paper Google File System](#)
- 2004, [Paper MapReduce](#) publicado por Google
- 2004, Doug Cutting trabajando en Google crea la base de Hadoop para Nutch
- 2006-2008, Doug Cutting se une a Yahoo (web search index)
- Marzo 2006, Apache Hadoop 0.1.0
  - Hadoop es el nombre del peluche de su hijo, un elefante amarillo





# Historia

- 2007-2008, [The New York Times](#) “convierte” 150 años de historia con Amazon EC2/S3 y Hadoop
  - TIFFs → PDF
    - 4TB + 1.5TB en S3
    - 11M artículos
    - 100 instancias
    - 24 horas
  - [Timemachine](#)
    - 405.000 TIFF+XML → PNG+JSON
    - 3,3M de artículos en SGML
    - 36 horas



# Historia

- 2009, 3 ingenieros crean Cloudera, más tarde se une Doug Cutting y otros ingenieros como Tom White
- 2010, Framework MapReduce algoritmos genéticos
- 2012, Informe Gartner 4.4M Jobs for Big Data
- 2014, A lot of Startups for Big Data
  - **ScalingData** 'Killer-app' (email, mensajería instantánea, videojuegos online)
  - **SNAP**, AmpLAB (Berkeley)



# SNAP, ampLab (Berkeley) 2014

- Secuenciador de alineamiento de ADN Open Source
  - <http://snap.cs.berkeley.edu/>
- Profesor David Patterson (Berkeley, ampLab)
  - RISC, RAID
- [Apache Spark](#) (alternativa MapReduce x100 in memory)
  - Integrado con Hadoop 2 YARN
- Caso de [Joshua Osborn](#)
  - En solo 90 min. aisló elementos del ADN
  - El 0.2% pertenecían a una bacteria extraña
- Búsqueda para cura contra el cáncer



# SNAP

SNAP is built on Spark, and it's already saving lives. Patterson spoke about a recent case where [a boy in Wisconsin was suffering from a mysterious illness](#) that had him trapped in a coma for weeks with brain swelling. He was sent to the University of California, San Francisco, where doctors worked with Patterson and AMPLab to process a sample of his DNA using SNAP. In about 90 minutes, the computer had isolated all the human elements, leaving just the .02 percent that wasn't. It belonged to a rare bacterium, which was treated immediately.

“How do you find the needle in the haystack?” Patterson asked. “Get rid of all the hay.”

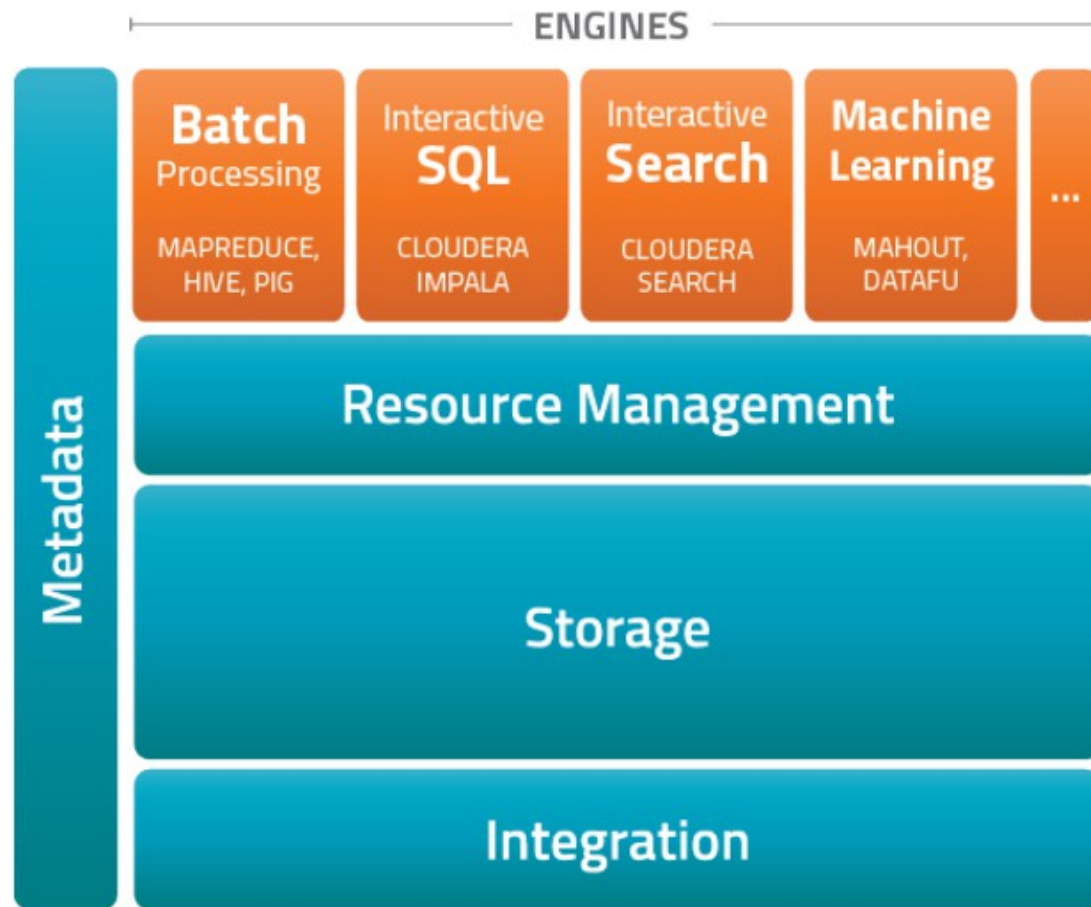


# Cloudera

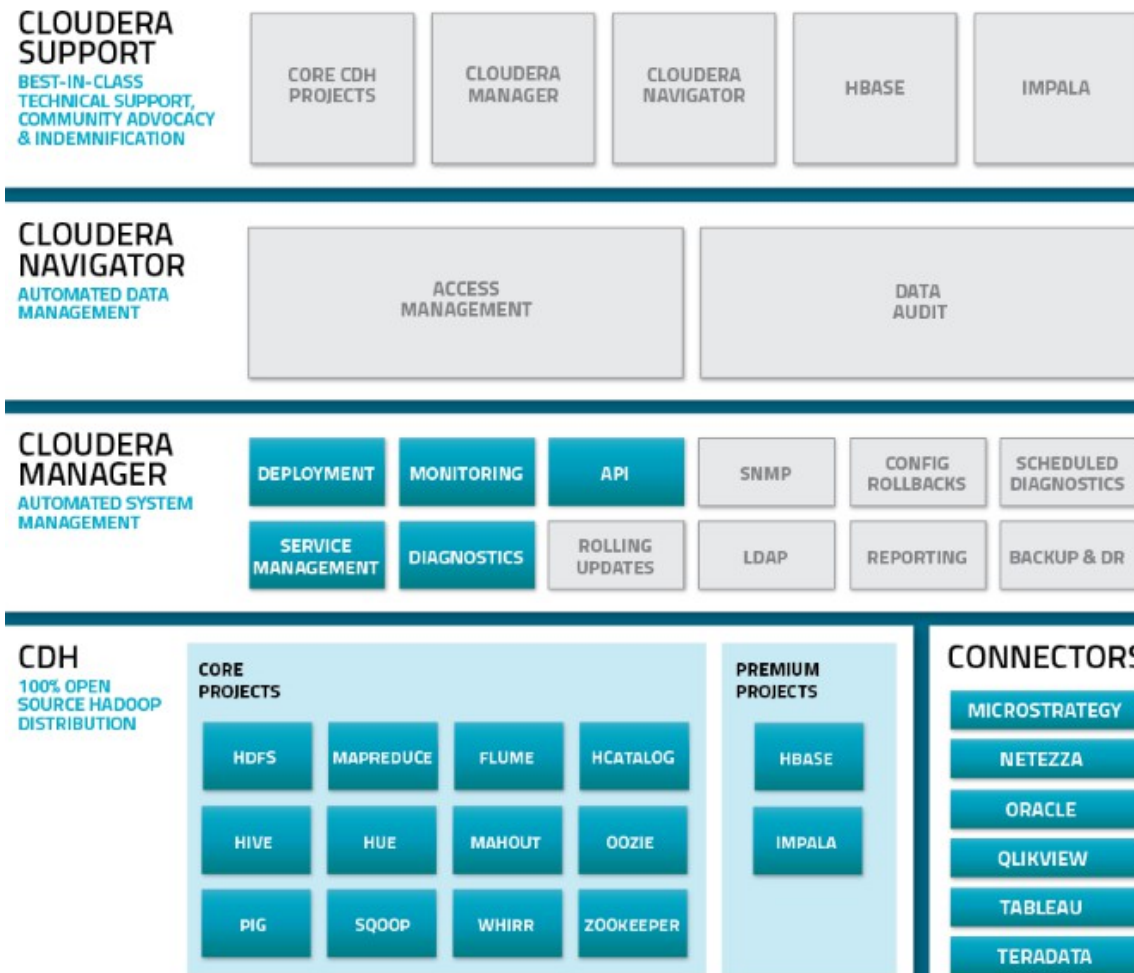
- Soporte de Apache Hadoop a empresas
- Certificaciones y cursos para Hadoop
- Modelo de negocio freemium
- CDH (Cloudera Distribution Including Apache Hadoop)
  - Distribución de Open Source
  - Paquetes RPM y Deb (Debian/Ubuntu/Suse)
  - Cloudera Manager



# Cloudera CDH



# Cloudera Standard



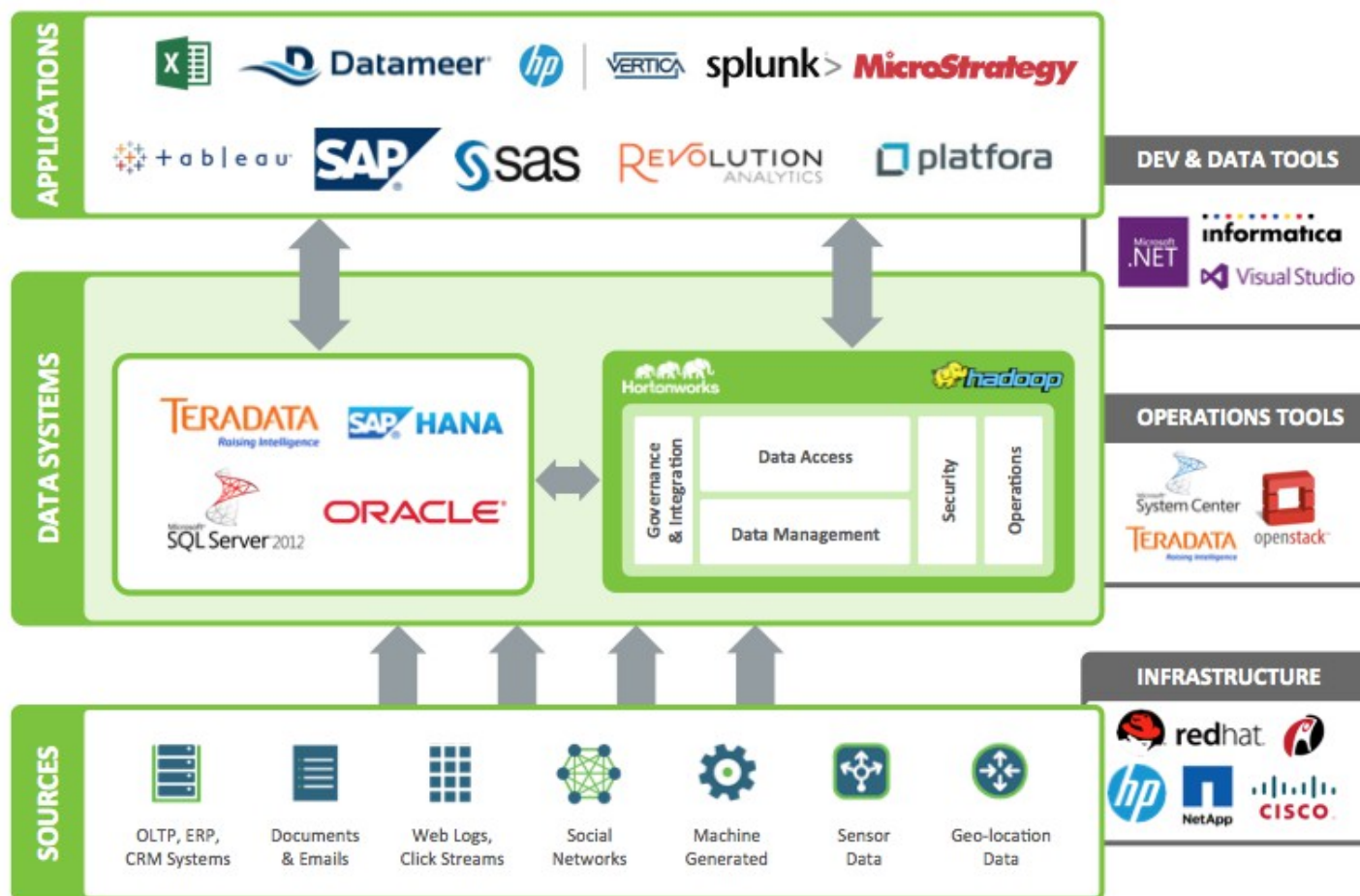
# HortonWorks

- Alternativa a Cloudera
- Fundada en 2011
- Servicio HDP análogo a CDH de Cloudera
- Certificaciones
- Partners como Yahoo, Microsoft, Red Hat o SAP
- Integración y soporte en muchas aplicaciones



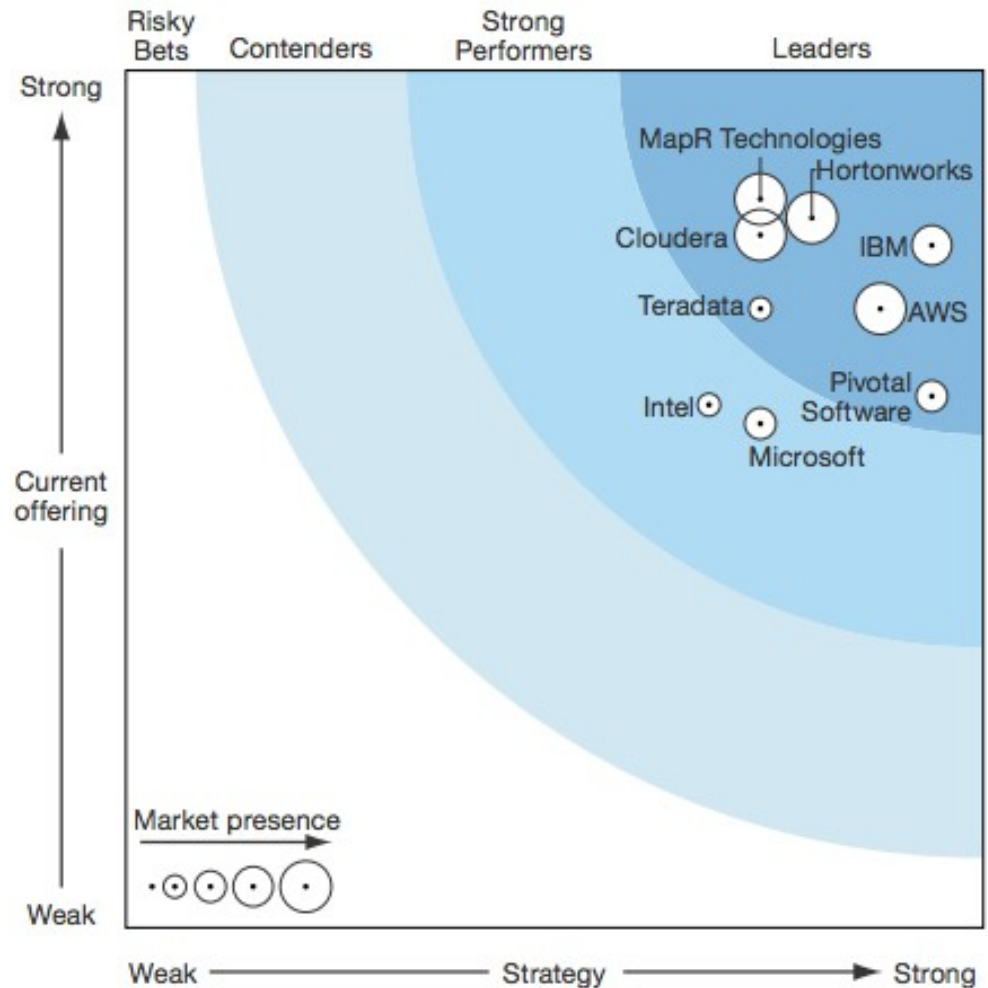


# HortonWorks



# Mercado

- Cloudera y HortonWorks: líderes de mercado
- Informe Forrester Research Q1 2014:
  - Big Data Hadoop Solutions

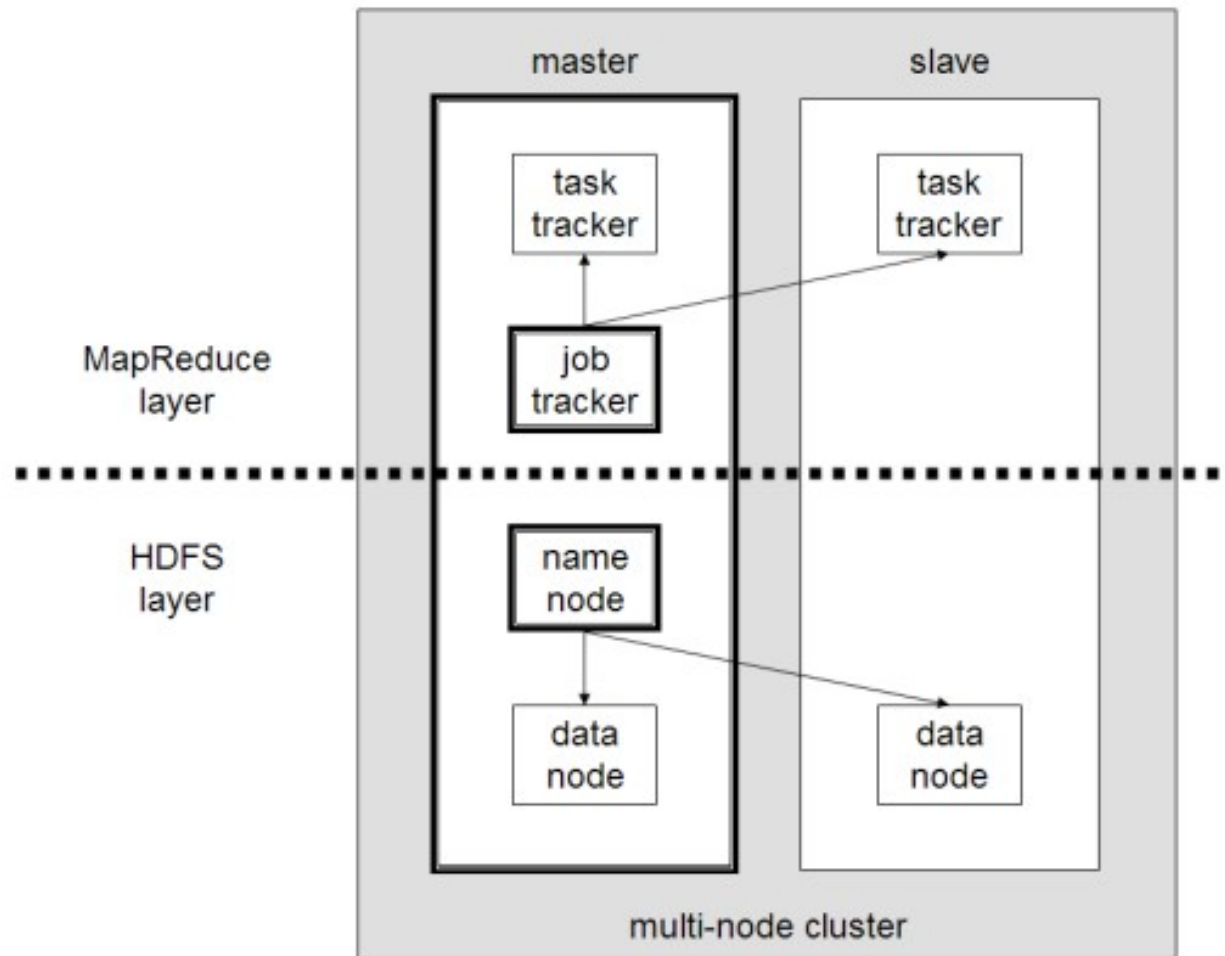


# Índice

- Introducción
- **Arquitectura**
- Requisitos
- Instalación
- HDFS
- MapReduce
- Ecosistema
- Alta Disponibilidad
- Resumen



# Arquitectura



# Arquitectura

- Almacenar y procesar los datos en los mismos nodos
- Planificación de trabajo “*data locality*”
  - Procesamiento en el mismo nodo donde están los datos
- Separación de los datos bloques (64-128 Mbytes)
  - 1000 discos (210MB/s) en paralelo
  - Lectura 3TB en 15 seg.
- Replicación
  - Mismo bloque replicado en 3 nodos



# Arquitectura

- Programación en Java al igual que Hadoop
  - Soporta otros lenguajes (Python, PHP, C, etc.)
- Simplificación del procesamiento
  - Trabajo  $\text{Map}(k1, v1) \rightarrow \text{list}(k2, v2)$
  - Recopilación intermedia y envío
  - Trabajo  $\text{Reduce}(k2, \text{list}(v2)) \rightarrow \text{list}(v3)$



# Componentes HDFS

- **NameNode**
  - Nodo maestro de HDFS
  - Almacena Metadatos
    - FSImage (checkpoint)
    - Edits (edit logs)
- **DataNode**
  - Nodos esclavos
  - Almacenan los bloques



# Componentes HDFS

## Secondary NameNode

- Liberar carga al NameNode
- Copia de respaldo de FSImage
  - Siempre en nodo alternativo a NameNode
- Aplica los últimos cambios en FSImage
  - Cada hora
  - Cada 64MBytes





# Componentes MapReduce

## JobTracker

- Gestor de los trabajos MapReduce
- Hasta 4 intentos por cada trabajo
  - En cada nodo que contiene el dato
- Registra el progreso de los trabajos

## Tasktracker

- Ejecuta un trabajo sobre un **bloque**
- Intenta lanzarse sobre el DataNode que tiene el dato



# Alta Disponibilidad

- **NameNode Active**
- **NameNode Standby**
  - Sustituye al SecondaryNameNode pero con HA
- **JournalNode**
  - Almacenan los metadatos de los NameNode
  - Al menos 1/3 deben estar levantados
- **ZooKeeper**
  - Cuidador del ecosistema de Hadoop
  - Habilitar Failover en HA
    - Por defecto es manual



# Índice

- Introducción
- Arquitectura
- **Requisitos**
- Instalación
- HDFS
- MapReduce
- Ecosistema
- Alta Disponibilidad
- Resumen



# Requisitos

- NameNode/Jobtracker
  - Requiere gran cantidad de memoria
    - 8-16 Gb
- DataNode/Tasktracker
  - 2-4 discos SAS/SSD
    - **¡Importante! No RAID**
  - Dual-Quad core
    - 2 tareas por Tasktracker



# Requisitos

- Imágenes VM de Cloudera
  - [http://www.cloudera.com/content/cloudera/en/downloads/quicksstart\\_vms/cdh-4-7-x.html](http://www.cloudera.com/content/cloudera/en/downloads/quicksstart_vms/cdh-4-7-x.html)
- VirtualBox, KVM o VMWare Player
  - En el taller usaremos VirtualBox
- OS 64-bit
  - 2 cores
  - 4Gb-8Gb RAM
  - 15Gb-30 Gb espacio libre en disco



## 4 VMs

- Configuración básica
  - 768MB-1536 MB de RAM
  - 1 core
- NAT en eth1
- Hostname: eth2 (internal network)
  - elephant: 192.168.0.1/24
  - tiger: 192.168.0.2/24
  - horse: 192.168.0.3/24
  - monkey: 192.168.0.4/24



# Importar VM base

Crear unha nova máquina virtual


## Nome da máquina virtual e tipo de sistema operativo

Insira un nome para a nova máquina virtual e seleccione o tipo de sistema operativo convidado que planea instalar na máquina virtual.

O nome da máquina virtual xeralmente indica a súa configuración de hardware e software. Será empregado por todos os compoñentes de VirtualBox para identificar a súa máquina virtual.

**Nome**

**Tipo de sistema operativo**

Sistema operativo: Linux 

Versión: Red Hat (64 bit)

< Volver Seguinte> Cancelar



# Importar VM base

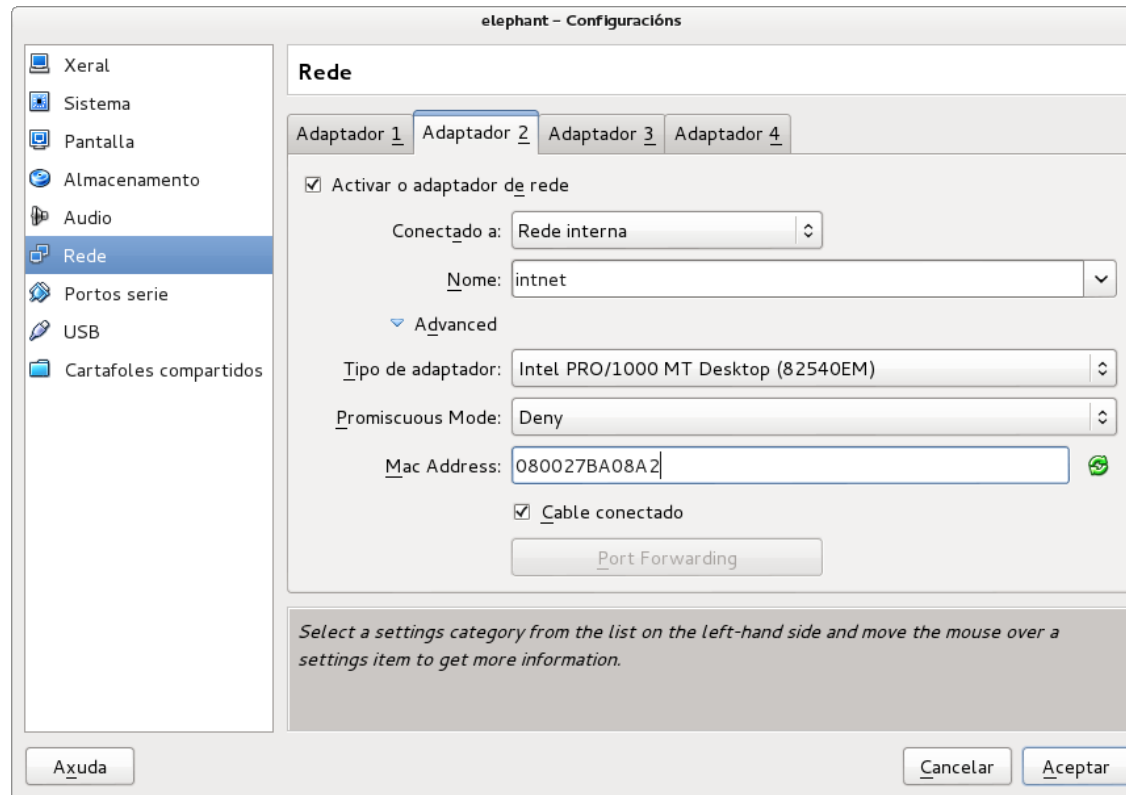




# Importar VM base



# Adaptador de Red



# Índice

- Introducción
- Arquitectura
- Requisitos
- **Instalación**
- HDFS
- MapReduce
- Ecosistema
- Alta Disponibilidad
- Resumen



# Instalación

- Configuración idioma teclado y zona horaria
- Actualización de paquetes
  - `$ yum update -y`
- Instalación de vi mejorado (*opcional*)
  - `$ yum install nano vim emacs -y`
- Detener los servicios de Cloudera Manager
  - Servicio web parando todos los procesos
  - Eliminar servicio en arranque



# Eliminar Cloudera Manager

The screenshot shows the Cloudera Manager web interface in a Mozilla Firefox browser. The address bar shows the URL `localhost.localdomain:7180/cm/services/status`. The page title is "All Services - Cloudera Manager - Mozilla Firefox". The Cloudera Manager logo and navigation tabs (Home, Services, Hosts, Activities, Diagnose, Audits, Charts, Administration) are visible. The "Services" tab is selected. The main content area shows "All Services" for "Cluster 1 - CDH4". A table lists the services and their roles. A context menu is open over the "flume1" service, showing options: "Add a Service", "Start", "Stop", "Deploy Client Configuration", "View Client Configuration URLs", "Rename Cluster", and "Delete". The "Stop" option is highlighted. A tooltip message says: "Stop all services in the cluster (Command Stop is not currently available for execution.)".

Name	Status	Role Counts
flume1	Stopped	1 Agent
hbase1	Stopped	1 RegionServer, 1 Master, 1 HBase Thrift Server
hdfs1	Stopped	1 SecondaryNameNode, 1 NameNode
hive1	Stopped	1 Hive Metastore Server, 1 Gateway
hue1	Stopped	1 Beeswax Server, 1 Hue Server
impala1	Stopped	1 Impala Catalog Server Daemon, 1 Impala Daemon, 1 Impala StateStore
ks_indexer1	Stopped	1 Lily HBase Indexer



# Eliminar pseudo-distribuido

- Cloudera Quickstart incluye paquetes pseudo-distribuidos
  - `$ sudo yum remove -y hadoop-0.20-conf-pseudo`
- Crear configuración base inicial
  - `$ sudo mv /etc/hadoop/conf /etc/hadoop/conf.alternatives`
  - `$ sudo mv /etc/hadoop/conf.empty /etc/hadoop/conf`
  - `$ touch /etc/hadoop/conf/hadoop-env.sh`



# Re-instalar paquetes

- Es necesario reinstalar los paquetes básicos
- Incluyen la configuración de ficheros y servicios
  - `$ sudo yum reinstall -y hadoop-hdfs-namenode hadoop-hdfs-datanode hadoop-0.20-mapreduce-tasktracker hadoop-hdfs-secondarynamenode hadoop-0.20-mapreduce-jobtracker`
- Habilitar arranque de los servicios
  - `$ for s in `cd /etc/init.d/; ls hadoop*`; do sudo chkconfig $s on; done`



# Configuración de Red

- Configurar tarjeta de red eth1
  - Configuración por defecto (NAT)
  - Gestionada por Network-Manager
    - */etc/sysconfig/network-scripts/ifcfg-eth1*

```
DEVICE="eth1"  
TYPE=Ethernet  
BOOTPROTO="dhcp"  
NM_CONTROLLED="yes"  
DEFROUTE=yes  
IPV4_FAILURE_FATAL=yes  
IPV6INIT=no  
NAME="eth1"  
ONBOOT=yes
```





# Configuración de Red

- Configurar tarjeta de red eth2
  - */etc/sysconfig/network-scripts/ifcfg-eth2*

```
DEVICE="eth2"  
TYPE=Ethernet  
BOOTPROTO="static"  
NM_CONTROLLED="no"  
IPADDR=192.168.1.1  
PREFIX=24  
DEFROUTE=yes  
IPV4_FAILURE_FATAL=yes  
IPV6INIT=no  
NAME="eth2"  
ONBOOT=yes
```



# Configuración de Red

- Configurar hosts del cluster
  - */etc/hosts*
    - *192.168.1.1 elephant*
    - *192.168.1.2 tiger*
    - *192.168.1.3 horse*
    - *192.168.1.4 monkey*
- Configurar hostname
  - */etc/sysconfig/network*
    - NETWORKING=yes
    - HOSTNAME=elephant



# Clonar VM

- Apagar elephant
  - System → Shut down
- Clonar desde VirtualBox
  - Machine → Clone...
  - Marcar “Reinitialize the MAC address of all network cards”
  - Full clone (copiar los discos)



# Índice

- Introducción
- Arquitectura
- Requisitos
- Instalación
- **HDFS**
- MapReduce
- Ecosistema
- Alta Disponibilidad
- Resumen

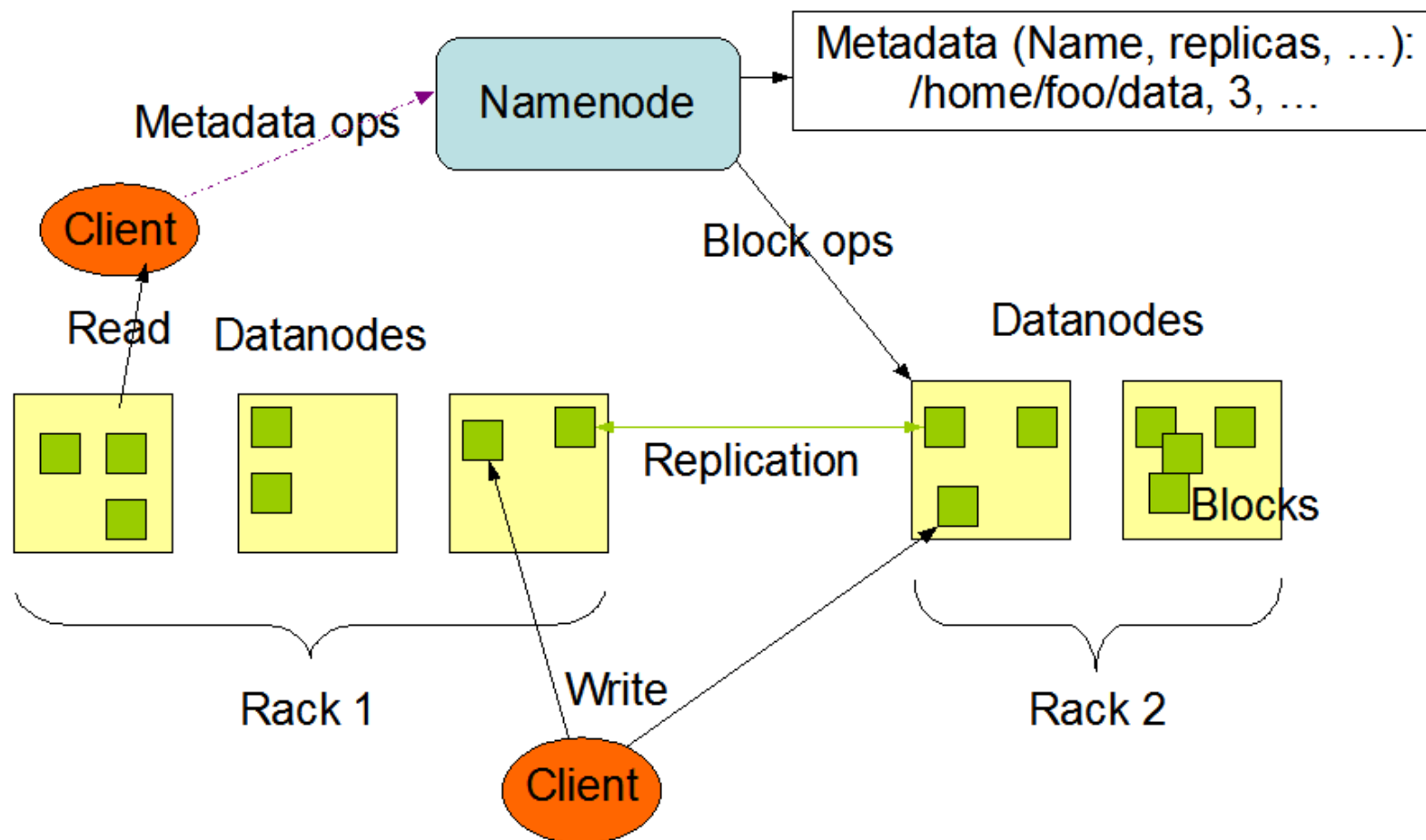


# HDFS (Hadoop Distributed FileSystem)

- Separa los datos en bloques (**64-128 Mbytes**)
- Acceso secuencial
  - Única escritura y lectura secuencial
  - Append (HBase)
- **NameNode** guarda los metadatos
  - Nombre fichero, permisos, fechas
  - Block-ids: bloques del fichero y que DN lo tienen
- **DataNode** almacena los bloques
  - Replicación x3



## HDFS Architecture



# Características

- *Data locality*
- Rack-aware
  - Replica dos bloques en otro mismo rack
- Switch-aware
  - x máquinas en *switches* distintos, réplicas en distintas LAN
- Anatomy File Read
  - Hadoop pide información al NameNode
  - El NN devuelve los DN en orden de cercanía



# NameNode

- Los bloques nunca pasan por el NameNode
  - Se transfieren entre DataNodes o al cliente
- **Memory Allocation**
  - Memoria suficiente en el NN para guardar los metadatos:
    - 150-200 bytes / ítem
    - 1 ítem / file metadata
    - 1 ítem / bloque
  - ¿Por qué Hadoop es óptimo para ficheros grandes?
    - Fichero 1Gb / 128Mb por bloque = 8 bloques = 9 ítems
    - Fichero 1Gb / 1Mb por bloque = 1025 ítems





# NameNode

- Si se cae, no hay acceso al HDFS
  - Hadoop soporta Federación de NameNode
    - Útil si no se tiene suficiente memoria física.
  - Almacena Metadatos
    - FSImage (checkpoint)
    - Edits (edit logs)
  - Opción recomendada Alta Disponibilidad
    - NameNode active y standBy
    - JournalNodes (al menos 3)
    - Failover automático con ZooKeeper



# DataNode

- Almacenamiento
  - Bloques de datos
  - Pueden ir en distintos discos
    - Sin RAID
  - Guarda un checksum del bloque
    - Verificación del bloque: lectura
- Envió *heartbeat* al NN para indicar disponibilidad
  - Si en 10 min. el NN no detecta un DN → *DN muerto*
    - NN solicita replicar los bloques del DN muerto
    - El bloque pasa de DN a DN, nunca por NN



# Configuración

- **core-site.xml**
  - **fs.default.name:** hdfs://elephant:8020
    - Identifica el NameNode
- **hdfs-site.xml**
  - **dfs.name.dir:** /disk1/dfs/nn,/disk2/dfs/nn
    - Ruta de los metadatos de NameNode
  - **dfs.data.dir:** /disk1/dfs/dn,/disk2/dfs/dn
    - Ruta de los datos de DataNode (más checksum)
  - **dfs.http.address:** elephant:50070
    - Servicio web información HDFS



# Otros parámetros

- **core-site.xml**
  - **hadoop.tmp.dir**
    - Ruta temporal, por defecto `/tmp/hadoop-${user-hadoop}`
- **hdfs-site.xml**
  - **dfs.block.size**
    - Tamaño bloque de datos, por defecto **64Mb**
  - **dfs.replication**
    - Número de bloques replicados, por defecto **3**
  - **dfs.datanode.du.reserved**
    - Espacio reservado por disco no ocupado por los bloques de HDFS en los DataNodes. Recomendable al menos 10Gb



# Iniciar servicios

- **elephant**
  - Namenode
    - # service hadoop-hdfs-namenode start
- **tiger**
  - Secondary NameNode
    - # service hadoop-hdfs-secondarynamenode start
- **All nodes**
  - DataNode
    - # service hadoop-hdfs-datanode start



# Índice

- Introducción
- Arquitectura
- Requisitos
- Instalación
- HDFS
- **MapReduce**
- Ecosistema
- Alta Disponibilidad
- Resumen

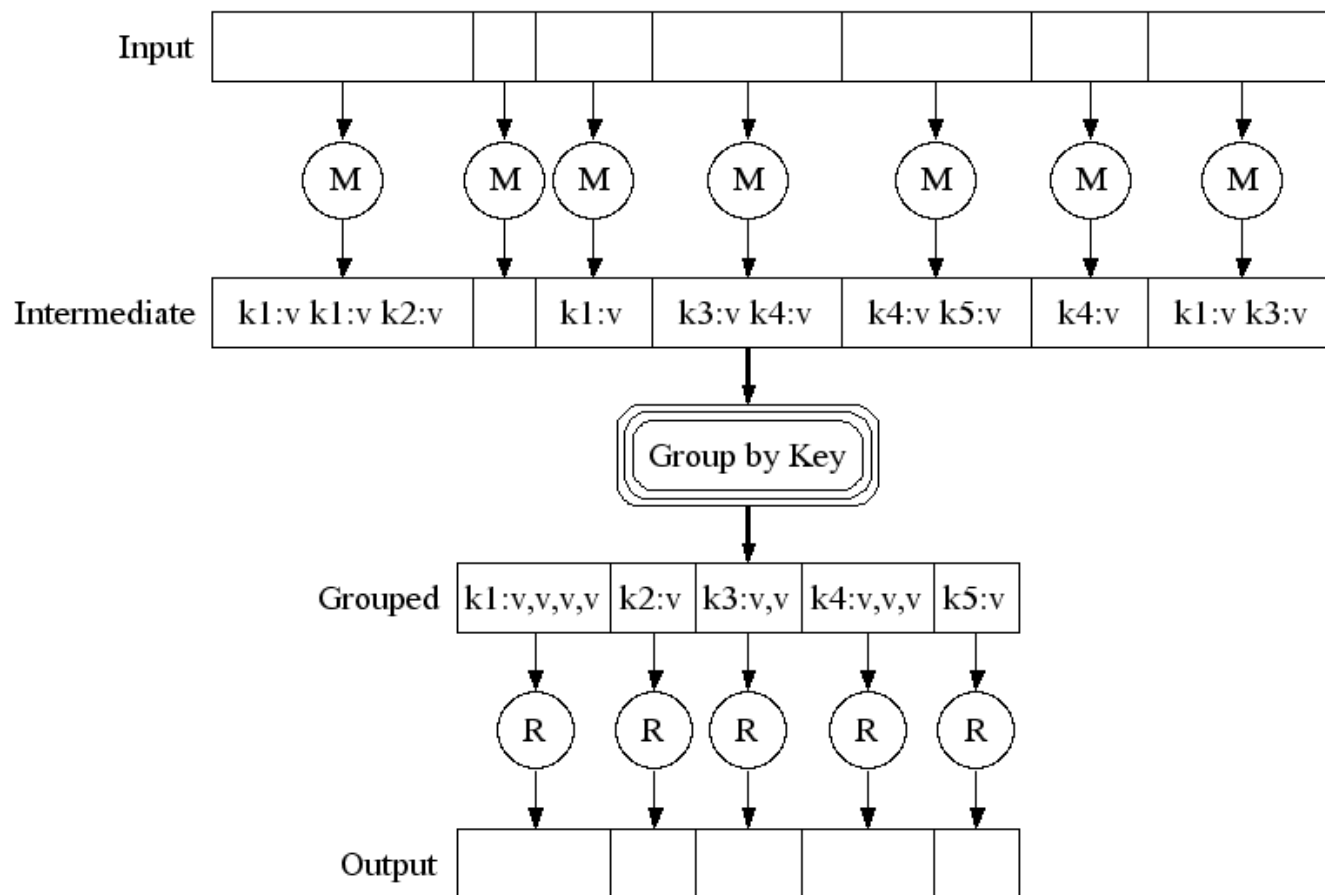


# Características

- Paralelización automática y distribuída
- Tolerancia a fallos
- Herramientas de monitorización
- Soporte para cualquier lenguaje con ***Hadoop Streaming***
- Los datos intermediarios se escriben en el disco local
- Terminología
  - **Job**: Mapper, Reducer y una lista de entradas
  - **Task**: Unidad de trabajo
  - Un Job se divide en Tasks



# MapReduce





# Fases

- **Mapper**
  - Input: datos de origen
  - Output: clave, valor
  - Tasks: Nº de bloques
- ***Shuffle & sort***
  - Reordenación de los datos salida de Map
- **Reduce**
  - Input: salida Mapper reordenados
  - Output: clave, valor resultado esperado
  - Tasks: Las define el desarrollador



# Componentes

- **JobTracker**

- Gestiona los trabajos MapReduce
- Si una tarea falla, se encarga de lanzarla de nuevo
- Soporta HA
  - JobTracker Activo y Standby

- **TaskTracker**

- Ejecuta las tareas MapReduce en un nodo
- Cada TaskTracker ejecuta por defecto 2 tareas
- Envía *heartbeats* al JobTracker



# Versiones

- **MRv1**
  - MapReduce clásica
  - Normalmente un solo JobTracker
  - Limitación de 1000 nodos para el JobTracker
- **MRv2 / YARN**
  - JobTracker → **Resource Manager**
  - TaskTracker → **NodeManager**
    - Actúa como JobTracker en el nodo
  - Ventaja: RM solo reparte la carga entre los distintos nodos



# Configuración

- **mapred-site.xml**
  - mapred.local.dir
    - Ruta local para fichero intermediarios
  - mapred.job.tracker
    - Especifica el JobTracker
  - mapred.system.dir
    - Ruta de ficheros compartidos durante la ejecución
  - mapreduce.jobtracker.staging.root.dir
    - Ficheros de configuración en HDFS



# Otros parámetros

- **mapred-site.xml**
  - mapred.tracktracker.map.task.maximum
    - Nº máximo de tareas Map (40-60)
  - mapred.tracktracker.reduce.task.maximum
    - Nº máximo de tareas Reduce ( $1.5 * \text{cores}$ )



# Iniciar servicios

- **horse**
  - JobTracker
    - # service hadoop-0.20-mapreduce-jobtracker start
- **All nodes**
  - TaskTracker
    - # service hadoop-0.20-mapreduce-tasktracker start

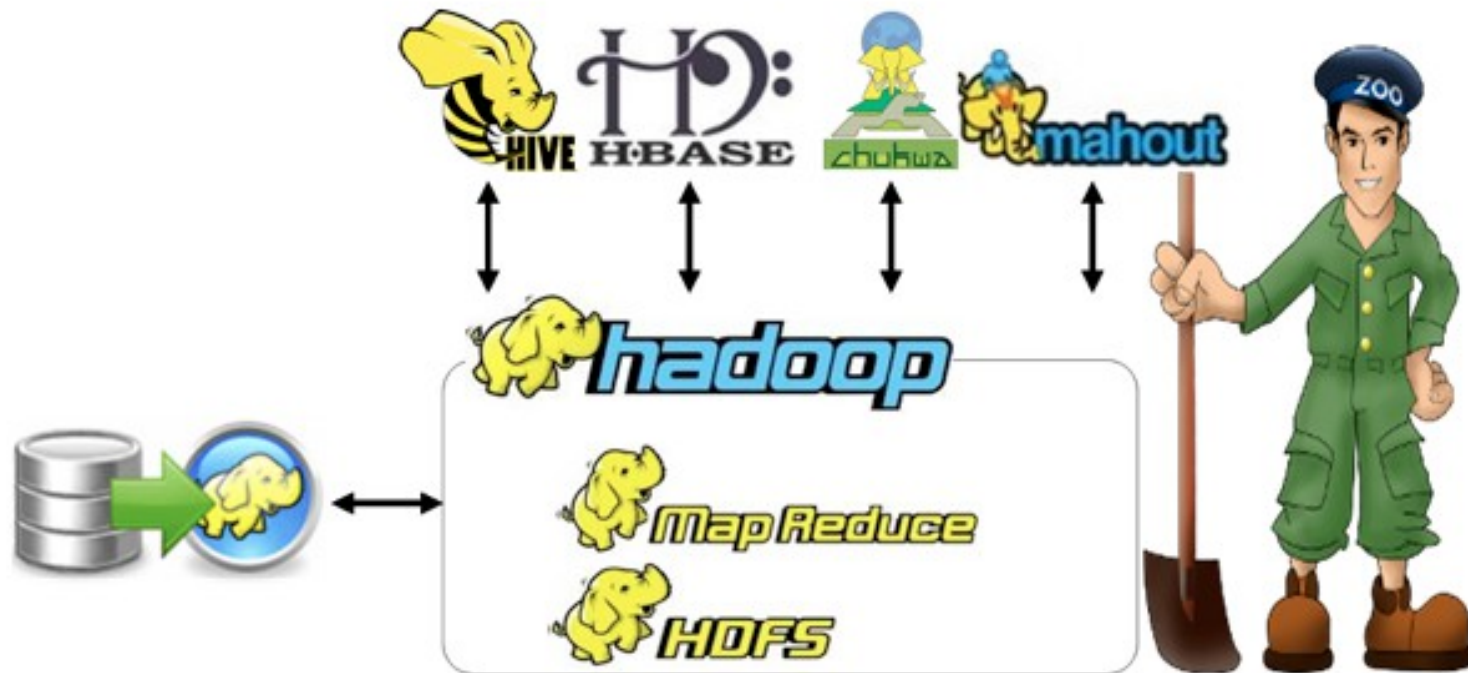


# Índice

- Introducción
- Arquitectura
- Requisitos
- Instalación
- HDFS
- MapReduce
- **Ecosistema**
- Alta Disponibilidad
- Resumen



# Ecosistema





# Ecosistema

- Data Integration
  - Flume, Sqoop
- Batch Processing
  - Hive, Pig
- Database NoSQL
  - HBase
    - NoSQL column-oriented
- Machine Learning & Data Mining
  - Mahout

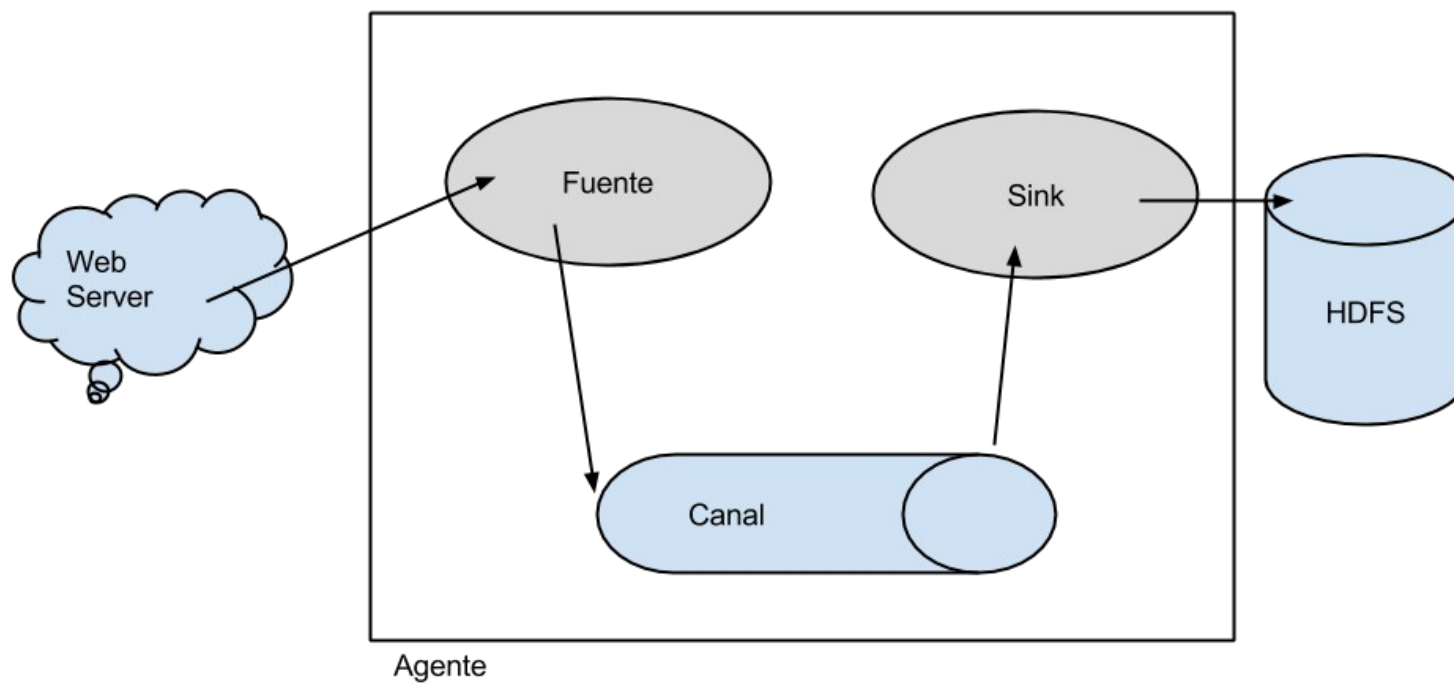


# Flume

- Integración de Datos
  - Cada servidor enviar sus logs a unos o varios agentes Flume
  - Soporta encriptado y compresión
- Agente
  - Almacena los datos en memoria
  - Envía a HDFS cuando alcanza un tamaño
- Canales de memoria
  - Establece tamaño caché
  - Comunica Agente con la fuente origen y el HDFS
    - Data Serialization Avro, Thift



# Flume



# Sqoop

- Integración de Datos
  - BD relacional ↔ HDFS
  - Soporte JDBC
    - Descargar el driver para el SGBD
      - Para MySQL `/usr/share/java/mysql-connector-java.jar`
    - Conectores alternativos (Netezza, Teradata, etc.)
- Sqoop ver. 2
  - Funciona como servicio
    - Disponible desde varios clientes
    - Requiere del driver JDBC en los clientes
    - Interfaz web: *WebHDFS* o *Http FS*



# Sqoop

- Instalación
  - `$ yum install -y sqoop`
  - `ls /usr/share/java/mysql-connector-java.jar`
- Importación tabla de MySQL
  - `$ sqoop import \`

`--connect jdbc:mysql://localhost/myDatabase \`

`--table myTable --fields-terminated-by '\t' \`

`--username myUser --password myPassword`

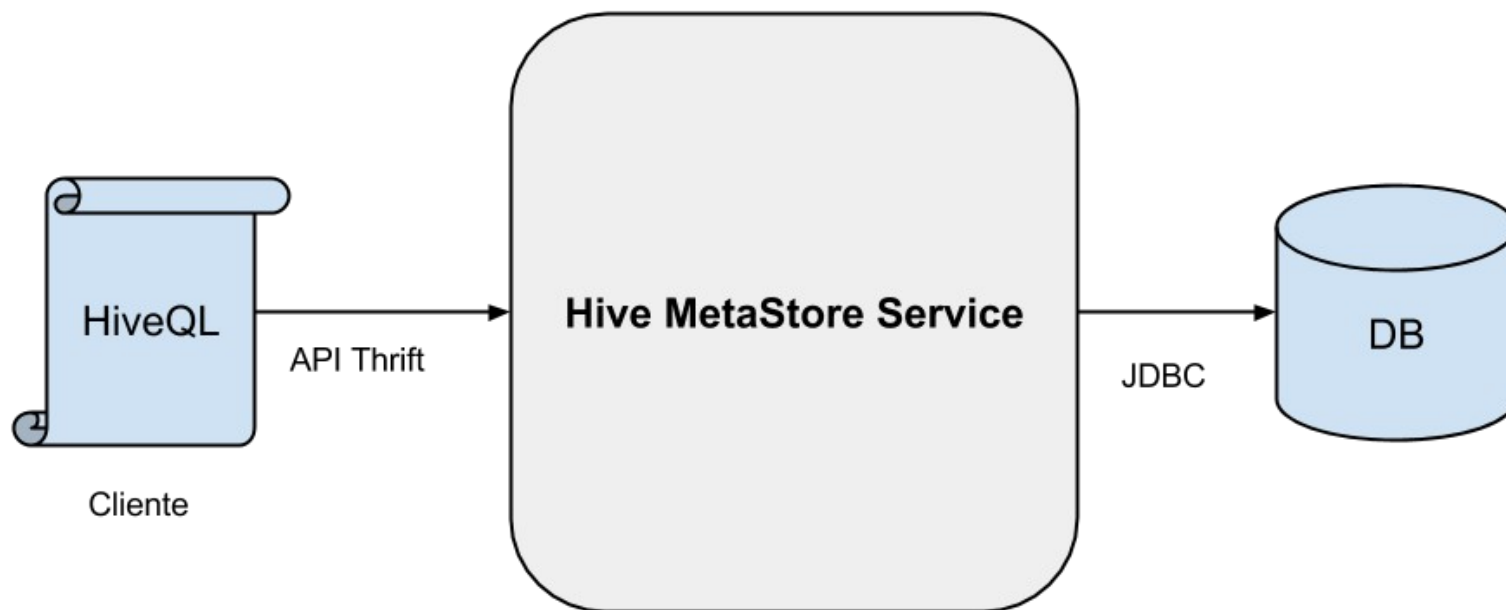


# Hive

- Desarrollado en Facebook para análisis de datos
- **HiveQL**
  - Consultas SQL sobre HDFS
    - Selects con Joins, Group by, Having, Order By
  - Lanza trabajos MapReduce
- **MetaStore**
  - Gestor de metadatos en DB relacional (MySQL, Derby)
  - MetaStore Local
  - Shared MetaStore
    - Múltiples usuarios ejecutan Hive compartido



# Hive



# Hive

- Instalación
  - `$ sudo yum install -y hive mysql-server libmysql-java`
  - Crear MetaStore
    - `mysql> create database metastore;`
    - Schema  
`/usr/lib/hive/scripts/metastore/upgrade/mysql/hive-schema-0.10.0.mysql.sql;`
  - Configurar Hive
    - `/etc/hive/conf/hive-site.xml`





# Pig

- Desarrollado por Yahoo! Para análisis de datos
- Alternativa a Hive
  - No usa sentencias SQL
- *Pig Latin*
  - Lenguaje de control de flujo
  - Define *bolsas* a través del intérprete *Grunt*
  - Soporta JOINS y filtros
  - Sentencia *STORE* almacena el resultado en HDFS



# Pig

- Instalación
  - `$ yum install -y pig`
- Ejemplo
  - `$ pig`
  - Definir bolsas:
    - `grunt> texts = LOAD 'binary_texts' AS (id_text:int, binary_file_id:int, english:chararray, spanish:chararray, created_date:chararray);`
    - `grunt> files = LOAD 'binary_files' AS (id_file:int, file:chararray, desc:chararray);`

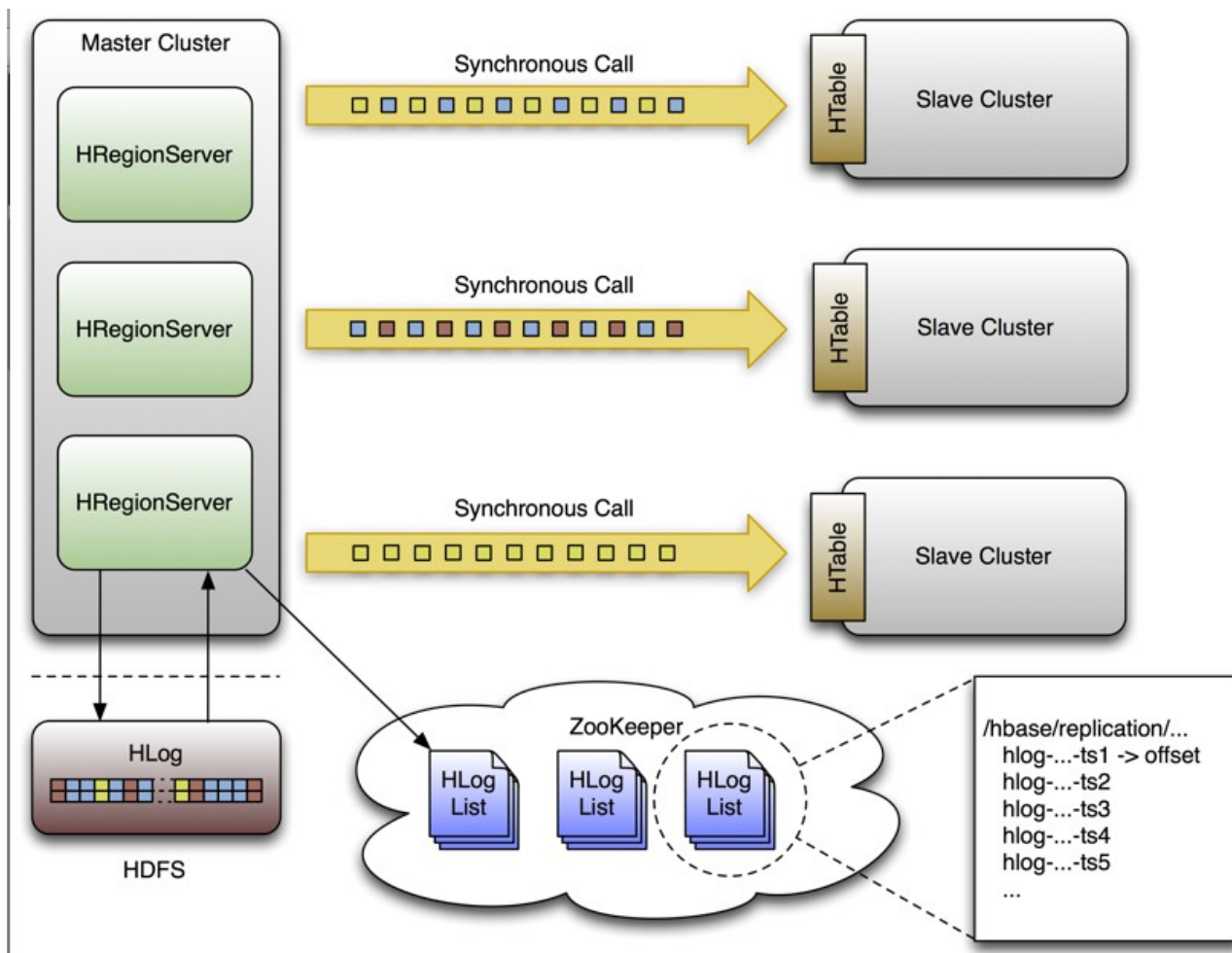


# HBase

- Bases de datos NoSQL
  - Orientado a columnas
- Sobre HDFS
  - Permite almacenar datos de poco tamaño
  - Operaciones APPEND
  - Familias de tablas
- Integración con Hive y Pig
- Acceso
  - hbase shell, Java API, Thrift, RESTful



# HBase



# HBase

- **ZooKeeper**
  - Servicio centralizado de alta disponibilidad
  - Gestiona el cluster de HBase
- **HBase Master**
  - Nodo maestro para el cliente Hbase
- **HBase RegionServer**
  - Recibe los datos y procesa las peticiones
  - Solo en nodos esclavos



# HBase

- Instalación
  - \$ sudo yum install -y hbase zookeeper-server hbase-master hbase-regionserver
- Configuración
  - Por defecto *Standalone*
  - */etc/hadoop/conf/hdfs-site.xml*
    - ***dfs.datanode.max.xcievers***
      - Límite ficheros que puede servir a la vez (4096)
    - ***hbase.cluster.distributed***
      - Habilitar HBase en modo distribuido
    - ***hbase.rootdir***
      - Ruta en HDFS de los datos de HBase



# HBase

- Ejemplos
  - Crear tabla test
    - *\$ hbase shell*
    - *hbase> create 'test', 'data'*
  - Insertar filas
    - *hbase> put 'test', 'row1', 'data:1', 'value1'*
  - Verificar tabla
    - *hbase> scan 'test'*
  - Obtener fila
    - *hbase> get 'test', 'row1'*



# Hue

- Contenedor web de aplicaciones de Hadoop
- Gestión de grupos y usuarios
- Hace más cómodo el uso de herramientas de Hadoop





# Hue

- Herramientas
  - Hive UI
  - Impala UI
  - File Browser
  - Job Browser
  - Job Designer
  - Oozie Workflow Editor
  - Shell UI



# Hue

- Instalación
  - `$ sudo yum install -y hue`
  - `sudo service hue start`
- Configuración
  - `/etc/hadoop/conf/hdfs-site.xml`
    - `dfs.webhdfs.enable`
      - Habilitar con “yes” el acceso HDFS
  - `/etc/hue/hue.ini`
    - `webhdfs_url=http://elephant:14000/webhdfs/v1/`
    - `jobtracker_host=horse`
    - `server_host=elephant`



# Mahout



- Herramienta para Machine-Learning
- Facilita la extracción de conocimiento
- Incluye algoritmos para **Data Mining**
  - Recomendación de objetos
  - Clustering o agrupamiento
  - Clasificación
  - Frecuencias de objetos



# Mahout

- Integración con Scala & Spark
  - **write one, run everywhere**
  - Alternativa distribuida para **R**
- Ejemplos
  - <http://mahout.apache.org/users/classification/breiman-example.html>



# Índice

- Introducción
- Arquitectura
- Requisitos
- Instalación
- HDFS
- MapReduce
- Ecosistema
- **Alta Disponibilidad**
- Resumen



# Alta Disponibilidad

- Si el NameNode cae
  - No se tiene acceso al clúster HDFS
  - SecondaryNameNode solo replica *FSImage*
- Si el JobTracker cae
  - No se pueden lanzar trabajos MapReduce

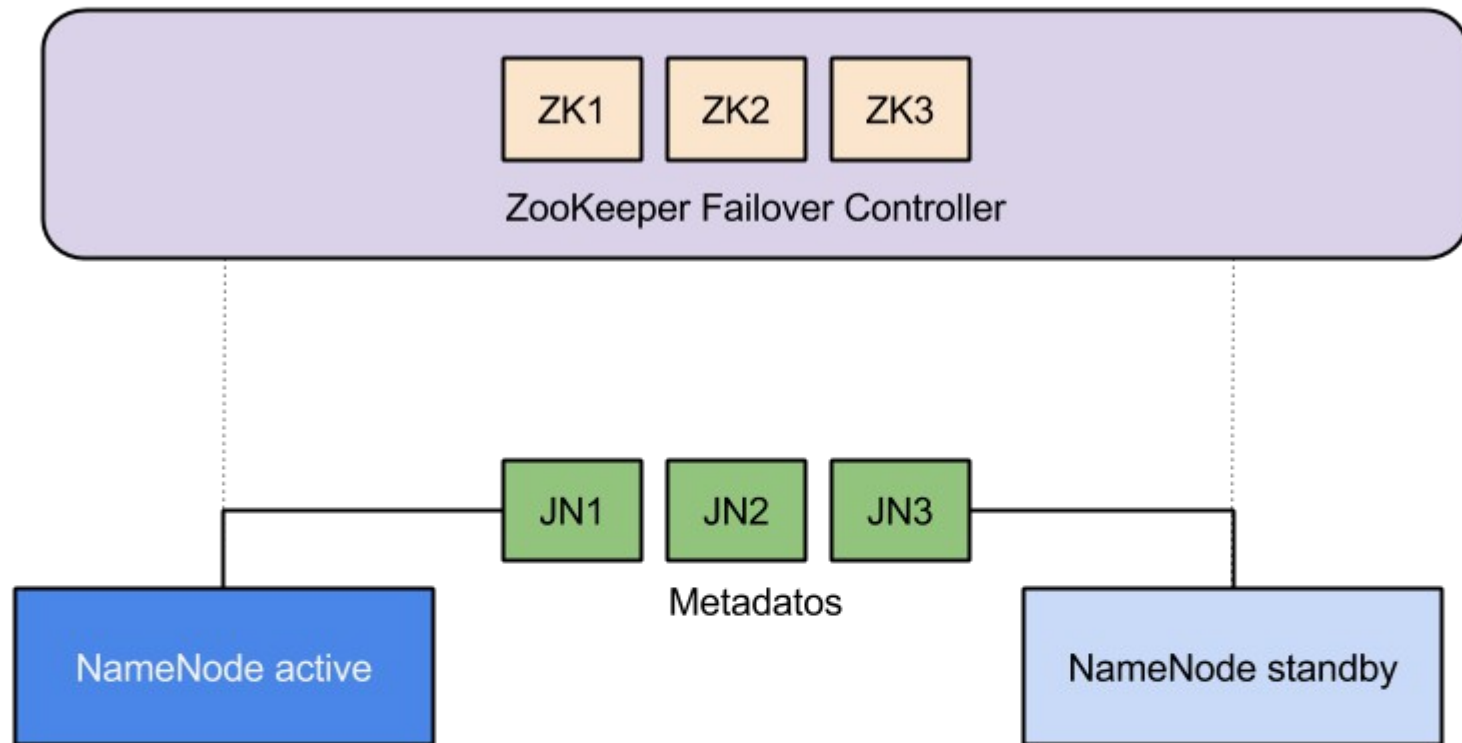


# Alta Disponibilidad

- NameNode with HA
  - NameNode Active
  - NameNode StandBy
    - SecondaryNameNode no se usa
- JournalNode
  - Almacena los metadatos
    - Los NN necesitan los metadatos actualizados **siempre**
  - Más de la mitad por cada NN
- ZooKeeper
  - Failover automático



# Alta Disponibilidad





# Resumen

- Hemos visto
  - Componentes básicos de Hadoop
  - Configuración en modo clúster
  - HDFS
  - Paradigma MapReduce
  - Algunas aplicaciones del ecosistema de Hadoop
  - Configuración alta disponibilidad



# Seguridad

- Cualquiera con acceso al clúster
  - Controlar el acceso a los datos
  - No hay autenticación de usuario
    - `$ sudo -u hdfs hadoop fs -ls -R /`
  - No soporta cifrado de disco en CDH4
- Soluciones
  - Aislar el clúster
  - Configurar Kerberos
    - <http://www.cloudera.com/content/cloudera/en/documentation/cdh4/latest/CDH4-Security-Guide/CDH4-Security-Guide.html>



# Referencias

- <http://hadoop.apache.org>
- <http://cutting.wordpress.com>
- <http://www.cloudera.com/content/cloudera/en/documentation/cdh4/latest/>
- <http://hortonworks.com>
- <http://www.gartner.com/newsroom/id/2207915>
- <https://gigaom.com/2013/03/04/the-history-of-hadoop-from-4-nodes-to-the-future-of-data/>
- <http://blogthinkbig.com/hadoop-open-source-big-data/>
- <https://gigaom.com/2014/08/02/the-lab-that-created-spark-wants-to-speed-up-everything-including-cures-for-cancer/>



# Referencias

- [http://wikibon.org/wiki/v/The\\_Hadoop\\_Wars:\\_Cloudera\\_and\\_Hortonworks%E2%80%99\\_Death\\_Match\\_for\\_Mindshare](http://wikibon.org/wiki/v/The_Hadoop_Wars:_Cloudera_and_Hortonworks%E2%80%99_Death_Match_for_Mindshare)
- <http://open.blogs.nytimes.com/2007/11/01/self-service-prorated-super-computing-fun/>
- <http://open.blogs.nytimes.com/2008/05/21/the-new-york-times-archives-amazon-web-services-timesmachine/>
- <http://www.norbertogallego.com/cloudera-punta-de-lanza-de-big-data/2013/07/19/>
- <http://www.cs.yale.edu/homes/tap/Files/hopper-wit.html>
- <http://research.google.com/archive/mapreduce-osdi04-slides/index-auto-0007.html>



# Referencias

- <https://forxa.mancomun.org/projects/mapreduceags/>
- <https://musicbrainz.org/>
- <http://db-engines.com/en/system/Cassandra%3BHBase%3BMongoDB>
- <http://www.bdisys.com/27/1/17/BIG%20DATA/HADOOP>
- <http://www.cloudera.com/content/cloudera/en/training/library/apache-hadoop-ecosystem.html>



# Referencias

- <http://gethue.com/>
- <http://pig.apache.org>
- <http://hive.apache.org>
- <http://sqoop.apache.org>
- <http://hbase.apache.org>
- <http://mahout.apache.org>
- <http://flume.apache.org>

