

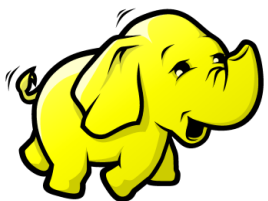


CURSOS DE VERANO 2014

APROXIMACIÓN PRÁCTICA A LA CIENCIA DE DATOS Y BIG DATA:
HERRAMIENTAS KNIME, R, HADOOP Y MAHOUT.

Entorno de Procesamiento Hadoop

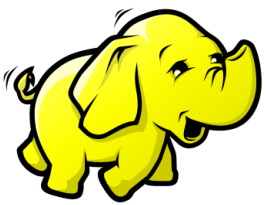
Sara Del Río García



¿Qué es Hadoop?

- Es un proyecto de código abierto escrito en Java administrado por la fundación Apache.
- Permite el almacenamiento y procesamiento distribuido de datos a gran escala en grandes clústeres de *commodity hardware*.
- Se inspiró en:
 - *Google's MapReduce*.
 - *Google's GFS (Google Distributed File system)*.



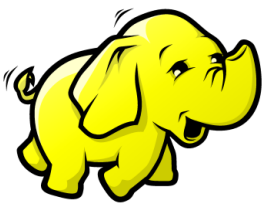


Características de Hadoop

- Consta de dos servicios principales:
 - **Almacenamiento:** HDFS.
 - **Procesamiento:** MapReduce.



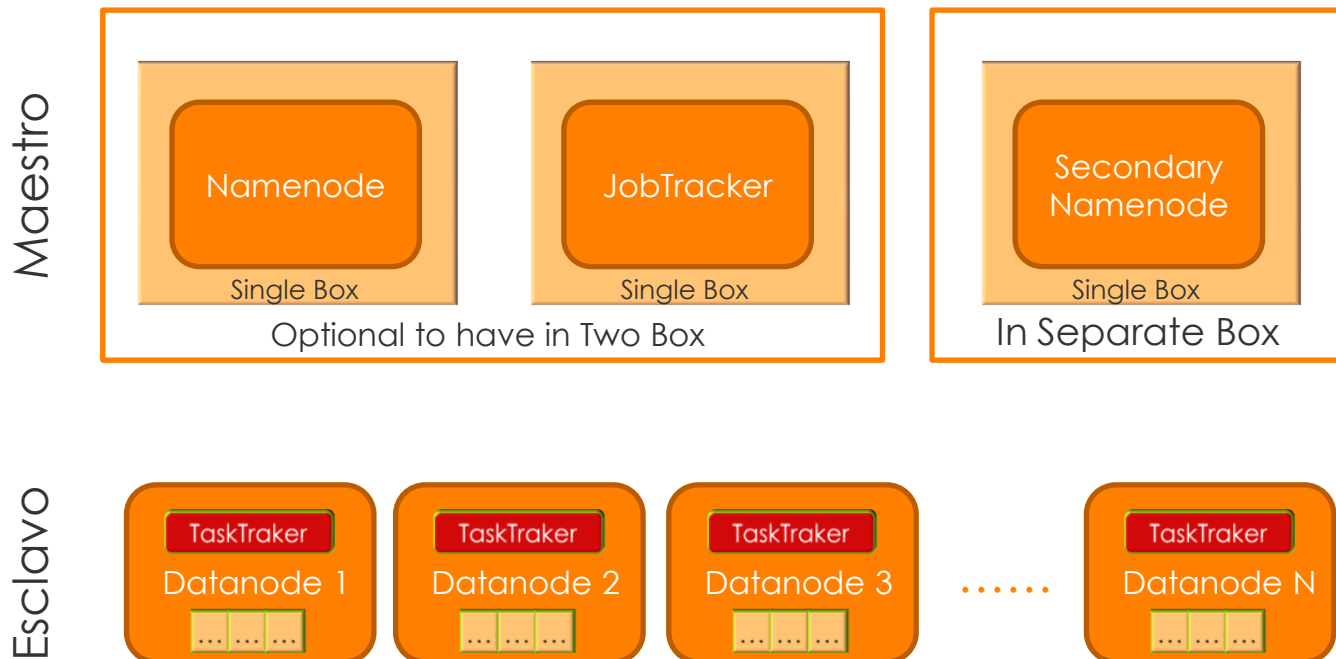
- Aporta una serie de ventajas:
 - **Bajo coste:** clústeres baratos / cloud.
 - **Facilidad de uso.**
 - **Tolerancia a fallos.**

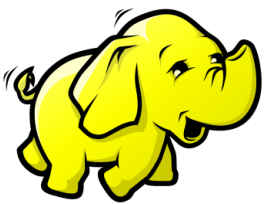


Arquitectura

Hadoop está diseñado para tener una **arquitectura Maestro Esclavo**.

- ▣ **Maestro:** NameNode, JobTracker
- ▣ **Esclavo:** {DataNode, TaskTraker}, ..., {DataNode, TaskTraker}

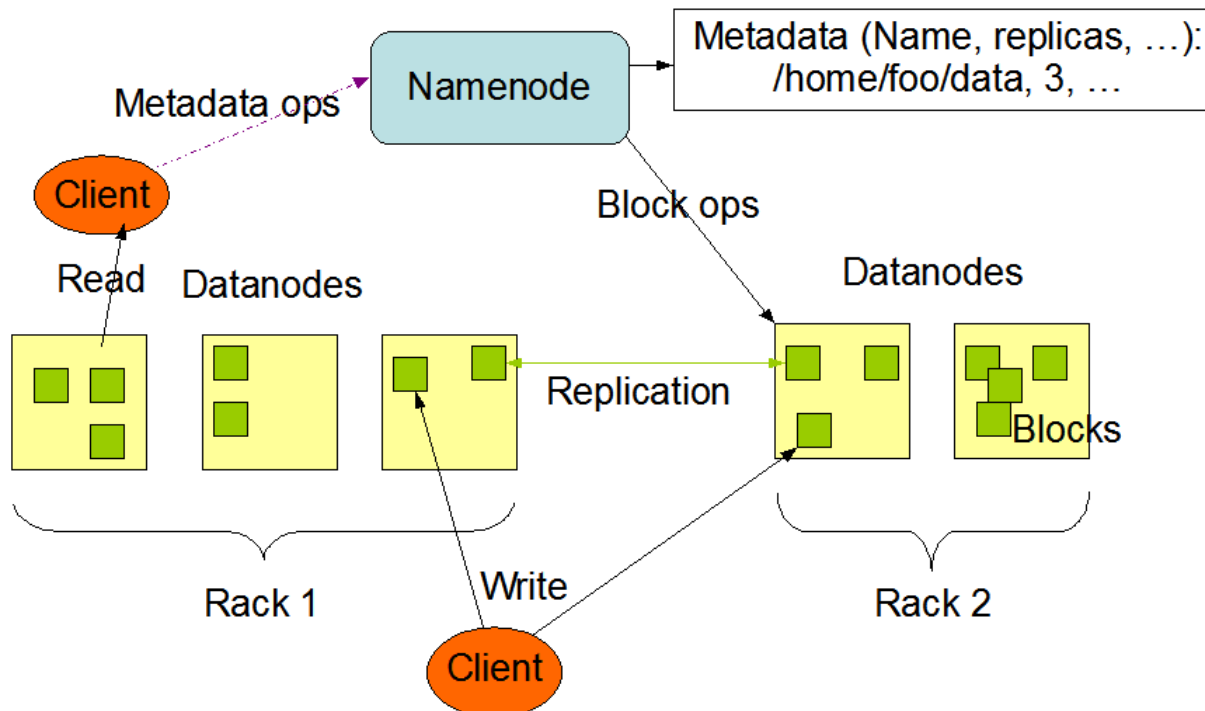


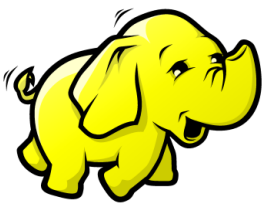


Arquitectura HDFS

- **Maestro:** NameNode
- **Esclavo:** {DataNode}, ..., {DataNode}

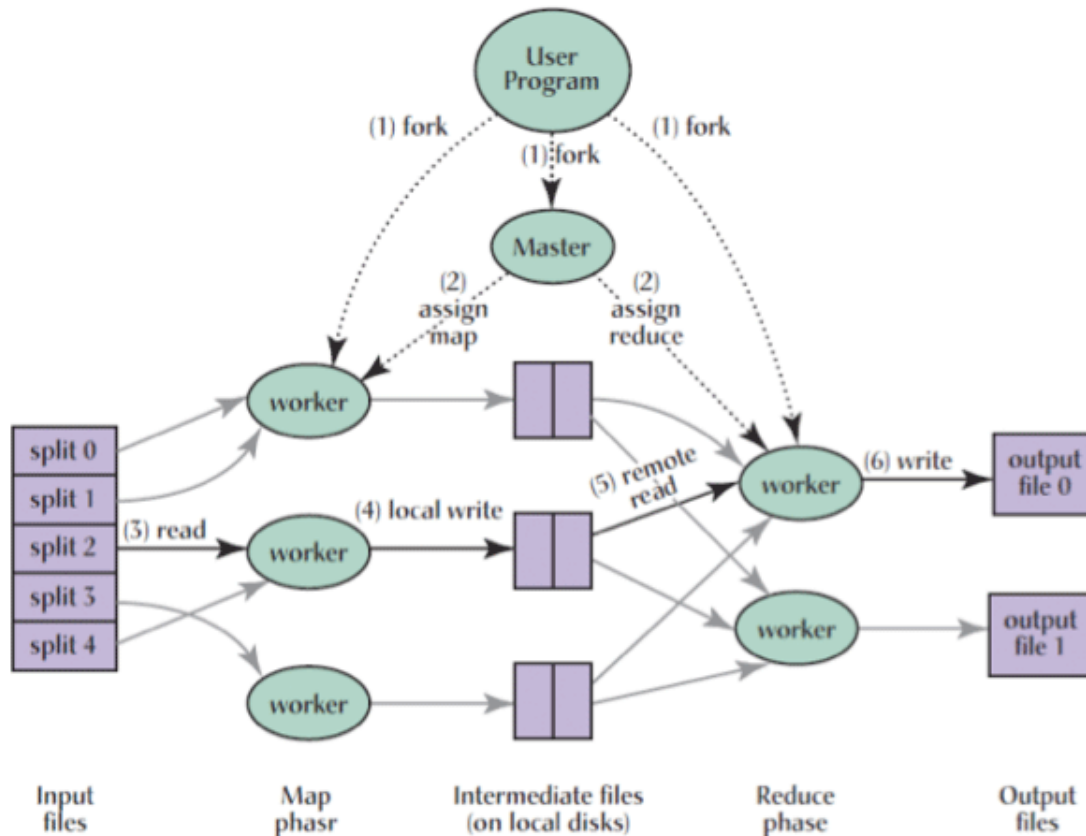
HDFS Architecture

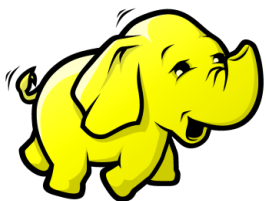




Arquitectura MapReduce

- Maestro : JobTracker
- Esclavo : {tasktraker},..., {Tasktraker}



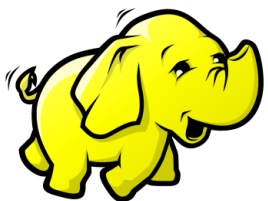


Ecosistema de Hadoop en la fundación Apache (I)

Algunas tecnologías relacionadas:

- **Hbase (la base de datos):** sistema de bases de datos NoSQL que corre sobre HDFS (inspirada en Google *BigTable*).
- **Cassandra:** Una base de datos escalable multi-master sin puntos individuales y fallo.
- **Hive (el data warehouse):** infraestructura de *data warehouse* construida sobre Hadoop.
- **Sqoop (la herramienta de ETL):** herramienta para transferencia eficiente de datos entre Hadoop y bases de datos relacionales.
- **Mahout (la plataforma de data mining):** algoritmos escalables de aprendizaje automático y minería de datos sobre Hadoop.





Ecosistema de Hadoop en la fundación Apache(II)



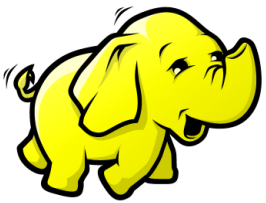
- **ZooKeeper (la herramienta de sincronización):** servicio centralizado de configuración, nombrado, sincronización distribuida y servicios de grupos para grandes sistemas distribuidos.

- **Chukwa:** Un sistema de recogida de datos para la gestión de grandes sistemas distribuidos.

- **Avro (el sistema de serialización):** una plataforma para codificar y homogeneizar los datos de forma que se puedan transmitir de forma óptima por la red.



- **Tez:** Sustituye al modelo “MapShuffleReduce” por un flujo de ejecución con grafos acíclico dirigido (DAG)

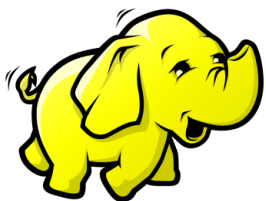


Ecosistema de Hadoop en la fundación Apache(III)



- **Giraph**: Procesamiento iterativo de grafos
- **Pig (el *helper* para analizar grandes volúmenes de datos)**: lenguaje de alto nivel (de flujo de datos) para facilitar la escritura de programas MapReduce.
- **Flume (el agregador de logs)**: capturar, analizar y monitorizar grandes ficheros de log.
- Y recientemente **Apache Spark**

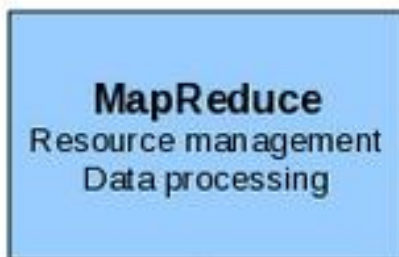




Evolución de Hadoop (I)

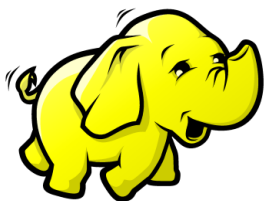
- **HDFS** es la capa de almacenamiento de datos para Hadoop y **MapReduce** era la capa de procesamiento de datos.
- Sin embargo, **MapReduce**, por sí solo, no es suficiente para la gran variedad de casos de uso a resolver.

Hadoop V1



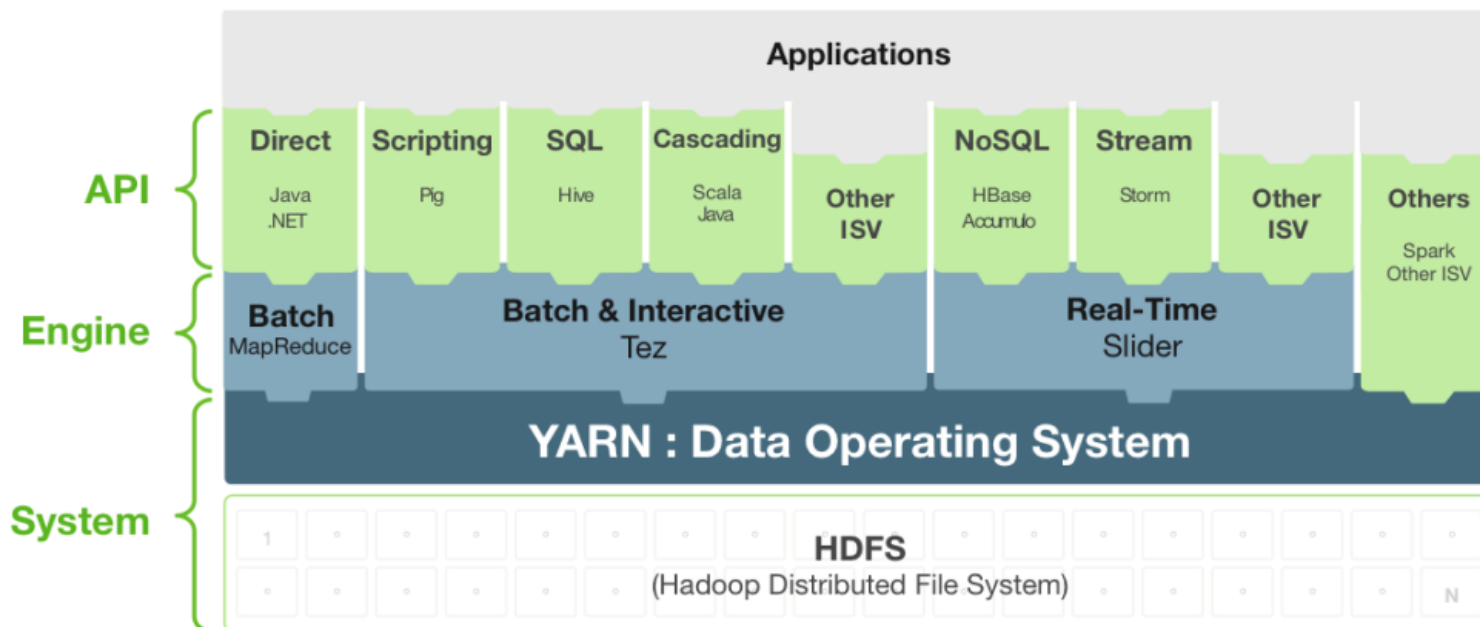
Hadoop V2



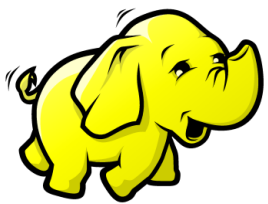


Evolución de Hadoop (II)

- Con **YARN (Yet Another Resource Negotiator)** Hadoop ahora tiene un entorno de **gestión de recursos y aplicaciones** distribuidas dónde se pueden implementar múltiples aplicaciones de procesamiento de datos totalmente personalizadas y específicas para realizar una tarea en cuestión.



- MapReduce** es, ahora, una de estas aplicaciones.



Ejemplos de algunas aplicaciones de MapReduce

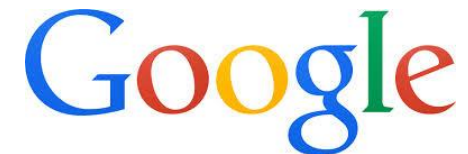
■ En Facebook

- Minería de Datos
- Detección de SPAM
- Gestión de Logs



■ En Google

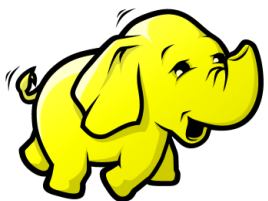
- Construcción de índices para el buscador (*pagerank*)
- Clustering de artículos en Google News
- Búsqueda de rutas en Google Maps



■ En I+D+i

- Análisis astronómico
- Bioinformática
- Simulación climática





¿Quien usa Hadoop?

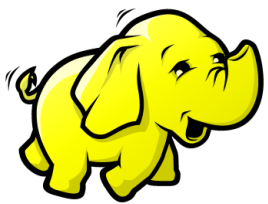
- eBay
- Facebook
- IBM
- The New York Times
- Tuenti
- Twitter

.... muchos más



The New York Times





<http://sortbenchmark.org/>

Primer hito de Hadoop: July 2008 - Hadoop Wins Terabyte Sort Benchmark

Uno de los grupos de Yahoo Hadoop ordenó 1 terabyte de datos en 209 segundos, superando el récord anterior de 297 segundos en la competición anual de ordenación de un terabyte (Daytona). Esta es la primera vez que un programa en Java de código abierto ganó la competición.

2008, 3.48 minutes

Hadoop

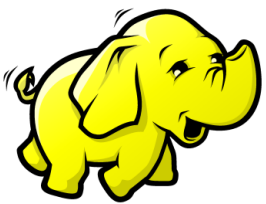
910 nodes x (4 dual-core processors, 4 disks, 8 GB memory)
Owen OMalley, Yahoo

2007, 4.95 min

TokuSampleSort

tx2500 disk cluster
400 nodes x (2 processors, 6-disk RAID, 8 GB memory)
Bradley C. Kuzmaul , MIT

Daytona	
Gray	2013, 1.42 TB/min
	Hadoop 102.5 TB in 4,328 seconds 2100 nodes x (2 2.3Ghz hexcore Xeon E5-2630, 64 GB memory, 12x3TB disks) Thomas Graves Yahoo! Inc.

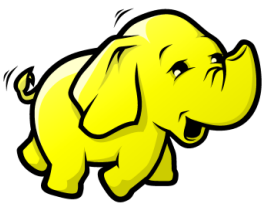


Hadoop Sorts a Petabyte in 16.25 Hours and a Terabyte in 62 Seconds (2009)

Bytes	Nodes	Maps	Reduces	Replication	Time
500,000,000,000	1406	8000	2600	1	59 seconds
1,000,000,000,000	1460	8000	2700	1	62 seconds
100,000,000,000,000	3452	190,000	10,000	2	173 minutes
1,000,000,000,000,000	3658	80,000	20,000	2	975 minutes

Yahoos' Hammer Cluster

- approximately 3800 nodes (in such a large cluster, nodes are always down)
- 2 quad core Xeons @ 2.5ghz per node
- 4 SATA disks per node
- 8G RAM per node (upgraded to 16GB before the petabyte sort)



Instalación en un clúster

Ejemplo ATLAS, infraestructura del grupo SCI²S

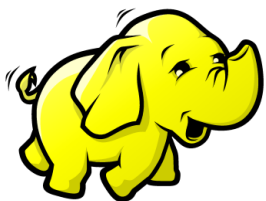


Cluster ATLAS: 4 super servers from Super Micro Computer Inc. (4 nodes per server)

Las características de cada nodo son:

- Microprocessors: 2 x Intel Xeon E5-2620 (6 cores/12 threads, 2 GHz, 15 MB Cache)
- RAM 64 GB DDR3 ECC 1600MHz, Registered
- 1 HDD SATA 1TB, 3Gb/s; (system)
- 1 HDD SATA 2TB, 3Gb/s; (distributed file system)



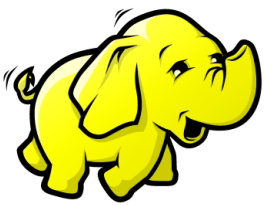


Instalación (I)

- Fuentes: <http://hadoop.apache.org/releases.html>
- Sistemas pre-configurados proporcionados por empresas. Las tres distribuciones más extendidas son:
 - **Cloudera** (www.cloudera.com): contribuidor activo al proyecto que proporciona una distribución comercial y no-comercial de Hadoop (CDH).
 - **MapR** (www.mapr.com).
 - **Hortonworks** (www.hortonworks.com).

Cada proveedor ofrece imágenes de VM con Linux y Hadoop ya instalado.





Instalación (II)

- Modos de funcionamiento:
 - *En un nodo:*
 - **Standalone (local):** de forma predeterminada, Hadoop está configurado para ejecutarse en un modo no distribuido, como un único proceso Java. Útil para la depuración.
 - **Pseudo-Distribuido:** cada demonio de Hadoop se ejecuta en un proceso Java independiente.
 - *En un clúster:*
 - **Distribuido.**

Instalación CDH4 con MRv1 en un único nodo Linux (I)

1. Asegurarse de tener Oracle JDK instalado:

1. `$ java -version`
2. `$ echo $JAVA_HOME`

2. Para instalar Oracle JDK (32 Bit) (Si no está instalado):

1. Extraer e instalar el contenido del archivo binario RPM:

1. `$ su`
2. `# cd /home/<user>`
3. `# chmod a+x jdk-6u45-linux-i586-rpm.bin`
4. `# ./jdk-6u45-linux-i586-rpm.bin`

2. Configurar la variable de entorno JAVA_HOME :

1. `# export JAVA_HOME=/usr/java/jdk1.6.0_45`
2. `# export PATH=$JAVA_HOME/bin:$PATH`
3. `# env | grep JAVA_HOME`

Donde <user> es el nombre de usuario Linux de cada usuario.

Instalación CDH4 con MRv1 en un único nodo Linux (II)

Instalación de CDH4 con con MRv1 en un único nodo bajo CentOS 6.5 en modo Pseudo-distribuido

1. Instalar el RPM para CDH4:

1. # `cd /home/<user>`
2. # `yum --nogpgcheck localinstall cloudera-cdh-4-0.i386.rpm`

2. Instalar Hadoop en modo Pseudo-distribuido:

1. # `yum install hadoop-0.20-conf-pseudo`

Donde <user> es el nombre de usuario Linux de cada usuario.

Instalación CDH4 con MRv1 en un único nodo Linux (III)

3. Iniciar Hadoop y verificar que funciona correctamente

Para MRv1, una instalación de Hadoop Pseudo-distribuida consiste en un nodo que ejecuta los cinco demonios Hadoop: NameNode, secondarynamenode, ResourceManager, DataNodes y NodeManager.

- ▣ Para ver los archivos:

```
# rpm -ql hadoop-0.20-conf-pseudo
```

```
/etc/hadoop/conf.pseudo
```

```
/etc/hadoop/conf.pseudo/README
```

```
/etc/hadoop/conf.pseudo/core-site.xml
```

```
/etc/hadoop/conf.pseudo/hadoop-env.sh
```

```
/etc/hadoop/conf.pseudo/hadoop-metrics.properties
```

```
/etc/hadoop/conf.pseudo/hdfs-site.xml
```

```
/etc/hadoop/conf.pseudo/log4j.properties
```

```
/etc/hadoop/conf.pseudo/mapred-site.xml
```

```
/etc/hadoop/conf.pseudo/yarn-site.xml
```

Instalación CDH4 con MRv1 en un único nodo Linux (IV)

3. Iniciar Hadoop y verificar que funciona correctamente

1. Formatear el NameNode: Antes de iniciar el NameNode por primera vez es necesario formatear el sistema de archivos.

```
# sudo -u hdfs hdfs namenode -format
```

2. Iniciar HDFS:

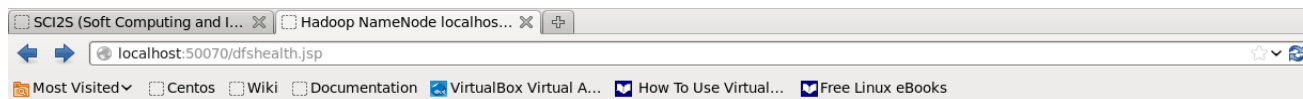
```
# for x in `cd /etc/init.d ; ls hadoop-hdfs-*`  
; do sudo service $x start ; done
```

Instalación CDH4 con MRv1 en un único nodo Linux (V)

3. Iniciar Hadoop y verificar que funciona correctamente

Podemos comprobar que los servicios se han iniciado correctamente accediendo a la consola web que proporciona el NameNode:

<http://localhost:50070/>



NameNode 'localhost.localdomain:8020' (active)

Started: Mon Aug 11 14:38:04 EDT 2014
Version: 2.0.0-cdh4.7.0, 8e266e052e423af592871e2dfe09d54c03f6a0e8
Compiled: Wed May 28 10:12:25 PDT 2014 by jenkins from (no branch)
Upgrades: There are no upgrades in progress.
Cluster ID: CID-3c321977-072b-4118-8176-c1e9b032e282
Block Pool ID: BP-564346478-127.0.0.1-1407782139900

[Browse the filesystem](#)
[NameNode Logs](#)

Cluster Summary

Security is **OFF**

1 files and directories, 0 blocks = 1 total.

Heap Memory used 34.51 MB is 67% of Committed Heap Memory 51.01 MB. Max Heap Memory is 966.69 MB.

Non Heap Memory used 17.71 MB is 99% of Committed Non Heap Memory 17.75 MB. Max Non Heap Memory is 96 MB.

Configured Capacity	:	27.55 GB
DFS Used	:	24 KB
Non DFS Used	:	3.18 GB
DFS Remaining	:	24.37 GB
DFS Used%	:	0.00%
DFS Remaining%	:	88.46%

Instalación CDH4 con MRv1 en un único nodo Linux (VI)

3. Iniciar Hadoop y verificar que funciona correctamente

3. Crear el directorio /tmp y establecer permisos:

1. # sudo -u hdfs hadoop fs -mkdir /tmp
2. # sudo -u hdfs hadoop fs -chmod -R 1777 /tmp

4. Crear los directorios del sistema MapReduce:

1. # sudo -u hdfs hadoop fs -mkdir -p
/var/lib/hadoop-
hdfs/cache/mapred/mapred/staging
2. # sudo -u hdfs hadoop fs -chmod 1777
/var/lib/hadoop-
hdfs/cache/mapred/mapred/staging
3. # sudo -u hdfs hadoop fs -chown -R mapred
/var/lib/hadoop-hdfs/cache/mapred

Instalación CDH4 con MRv1 en un único nodo Linux (VII)

3. Iniciar Hadoop y verificar que funciona correctamente
5. Verificar la estructura de ficheros HDFS:

```
# sudo -u hdfs hadoop fs -ls -R /
```

Se mostrará lo siguiente:

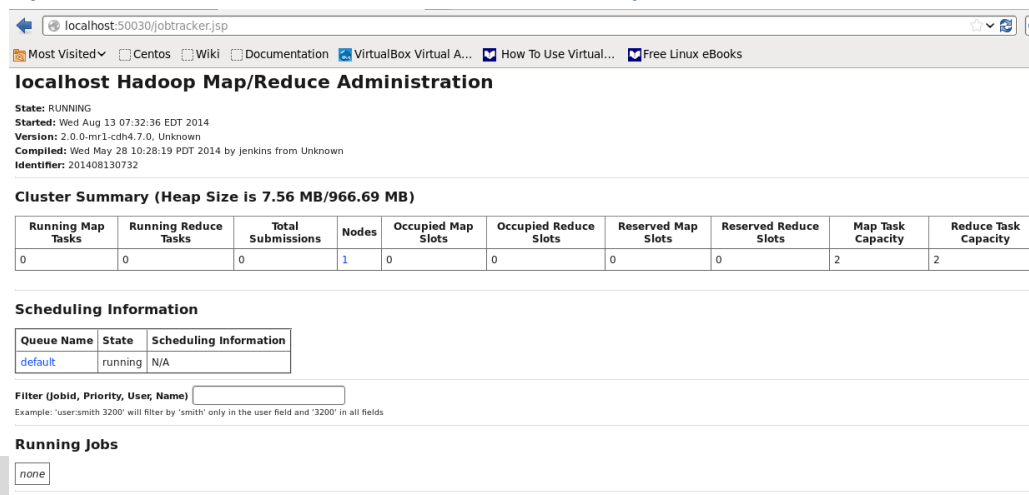
```
drwxrwxrwt - hdfs supergroup 0 2012-04-19 15:14 /tmp
drwxr-xr-x - hdfs supergroup 0 2012-04-19 15:16 /var
drwxr-xr-x - hdfs supergroup 0 2012-04-19 15:16 /var/lib
drwxr-xr-x - hdfs supergroup 0 2012-04-19 15:16 /var/lib/hadoop-hdfs
drwxr-xr-x - hdfs supergroup 0 2012-04-19 15:16 /var/lib/hadoop-hdfs/cache
drwxr-xr-x - mapred supergroup 0 2012-04-19 15:19 /var/lib/hadoop-hdfs/cache/mapred
drwxr-xr-x - mapred supergroup 0 2012-04-19 15:29 /var/lib/hadoop-hdfs/cache/mapred/mapred
drwxrwxrwt - mapred supergroup 0 2012-04-19 15:33 /var/lib/hadoop-hdfs/cache/mapred/mapred/staging
```

Instalación CDH4 con MRv1 en un único nodo Linux (VIII)

3. Iniciar Hadoop y verificar que funciona correctamente
6. Iniciar MapReduce

```
# for x in `cd /etc/init.d ; ls hadoop-0.20-  
mapreduce-*` ; do sudo service $x start ; done
```

Podemos comprobar que los servicios se han iniciado correctamente accediendo a la consola web que proporciona el JobTracker : <http://localhost:50030/>



The screenshot shows the web interface for the Localhost Hadoop Map/Reduce Administration. The page title is "localhost:50030/jobtracker.jsp". The status is "RUNNING". The version is "2.0.0-mr1-cdh4.7.0, Unknown". The cluster summary shows 1 node, 0 occupied map slots, 0 occupied reduce slots, 0 reserved map slots, and 0 reserved reduce slots. The scheduling information shows the default queue is running. The running jobs section shows no jobs.

Running Map Tasks	Running Reduce Tasks	Total Submissions	Nodes	Occupied Map Slots	Occupied Reduce Slots	Reserved Map Slots	Reserved Reduce Slots	Map Task Capacity	Reduce Task Capacity
0	0	0	1	0	0	0	0	2	2

Queue Name	State	Scheduling Information
default	running	N/A

Filter (Jobid, Priority, User, Name)

Example: 'usersmith 3200' will filter by 'smith' only in the user field and '3200' in all fields

Running Jobs
none

Instalación CDH4 con MRv1 en un único nodo Linux (IX)

3. Iniciar Hadoop y verificar que funciona correctamente
 7. Crear los directorios de usuario

```
# sudo -u hdfs hadoop fs -mkdir /user/<user>
# sudo -u hdfs hadoop fs -chown <user>
  /user/<user>
```

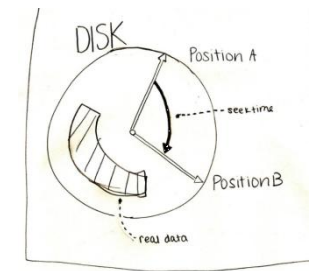
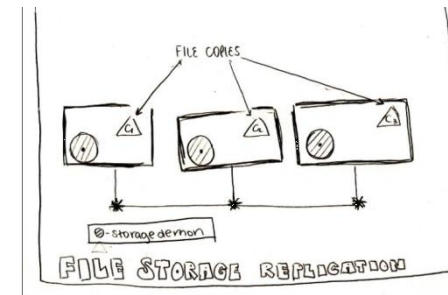
Donde <user> es el nombre de usuario Linux de cada usuario.

HDFS: *Hadoop Distributed File System*

- Hadoop puede acceder a diferentes tipos de sistemas de ficheros (local, HDFS, KFS, ...). No obstante se recomienda el uso de HDFS.

- Entre sus **ventajas** destacan:
 - Diseñado para almacenar ficheros muy grandes en *commodity hardware*.
 - Elevado ancho de banda.
 - Fiabilidad mediante replicación.

- También tiene algunos **inconvenientes**:
 - Elevada latencia.
 - Poco eficiente con muchos ficheros pequeños.



HDFS: *Hadoop Distributed File System*

HDFS Cuenta con tres interfaces:

- ▣ **API de programación.**
- ▣ **Interfaz web:**
 - ▣ Puerto 50070 del Namenode.
- ▣ **Línea de comandos:**
 - ▣ HDFS tiene su propia shell.
 - ▣ Ayuda: `$ hadoop fs -help`

HDFS: *Hadoop Distributed File System*

Algunos comandos:

Comandos	Descripción
<code>hadoop fs -ls <path></code>	Lista ficheros
<code>hadoop fs -cp <src> <dst></code>	Copia ficheros de HDFS a HDFS
<code>hadoop fs -mv <src> <dst></code>	Mueve ficheros de HDFS a HDFS
<code>hadoop fs -rm <path></code>	Borra ficheros en HDFS
<code>hadoop fs -rmr <path></code>	Borra recursivamente
<code>hadoop fs -cat <path></code>	Muestra fichero en HDFS
<code>hadoop fs -mkdir <path></code>	Crea directorio en HDFS
<code>hadoop fs -put <localsrc> <dst></code>	Copia ficheros de local a HDFS
<code>hadoop fs -copyToLocal <src> <localdst></code>	Copia ficheros de HDFS a local

Referencias

- **CDH4 Quick Start Guide:**

<http://www.cloudera.com/content/cloudera-content/cloudera-docs/CDH4/latest/CDH4-Quick-Start/CDH4-Quick-Start.html>

- **Hadoop:**

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

Happy Hadooping!

