



Universidad Autónoma
de Madrid

Campus Internacional
excelencia
UAM
CSIC



Escuela Politécnica Superior

Arquitecturas para tratar grandes volúmenes de información

Procesamiento de Datos a Gran Escala

Cluster Hadoop

- Introducción a Hadoop
- Hadoop se puede instalar de tres maneras distintas:
 - Standalone
 - Pseudo-Distributed
 - Fully Distributed

Google Origins

The Google File System

Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung
2003
Google



MapReduce: Simplified Data Processing on Large Clusters

2004
Jeffrey Dean and Sanjay Ghemawat
jeff@google.com, sanjay@google.com
Google, Inc.

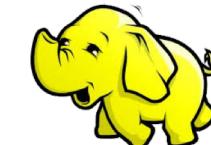


2006
Bigtable: A Distributed Storage System for Structured Data
Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach
Mike Burrows, Tushar Chandra, Andrew Fikes, Robert E. Gruber
(fay,jeff,sanjay,wilsonk,deborah,mike,tushar,afikes,gruber@google.com)
Google, Inc.



Abstract
Bigtable is a distributed storage system for managing structured data that is designed to scale to a very large petabytes of data across thousands of commodity servers. Many projects at Google store data in Bigtable, including web indexing, Google Earth, and Google Finance. These applications place very different demands on Bigtable, both in terms of data size (from URLs to

achieved scalability and high performance, but Bigtable provides a different interface than such systems. Bigtable does not support a full relational data model; instead, it provides clients with a simple data model that supports dynamic control over data layout and format, and allows clients to reason about the locality properties of data represented in the underlying storage. Data is indexed using row and column names that can be arbitrary strings. Bigtable also treats data as uninterpreted str-



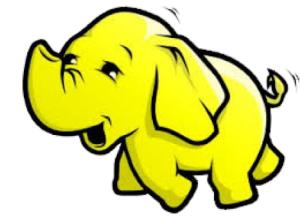
What is Hadoop?

- **Hadoop:**

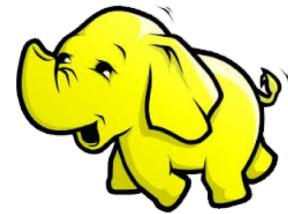
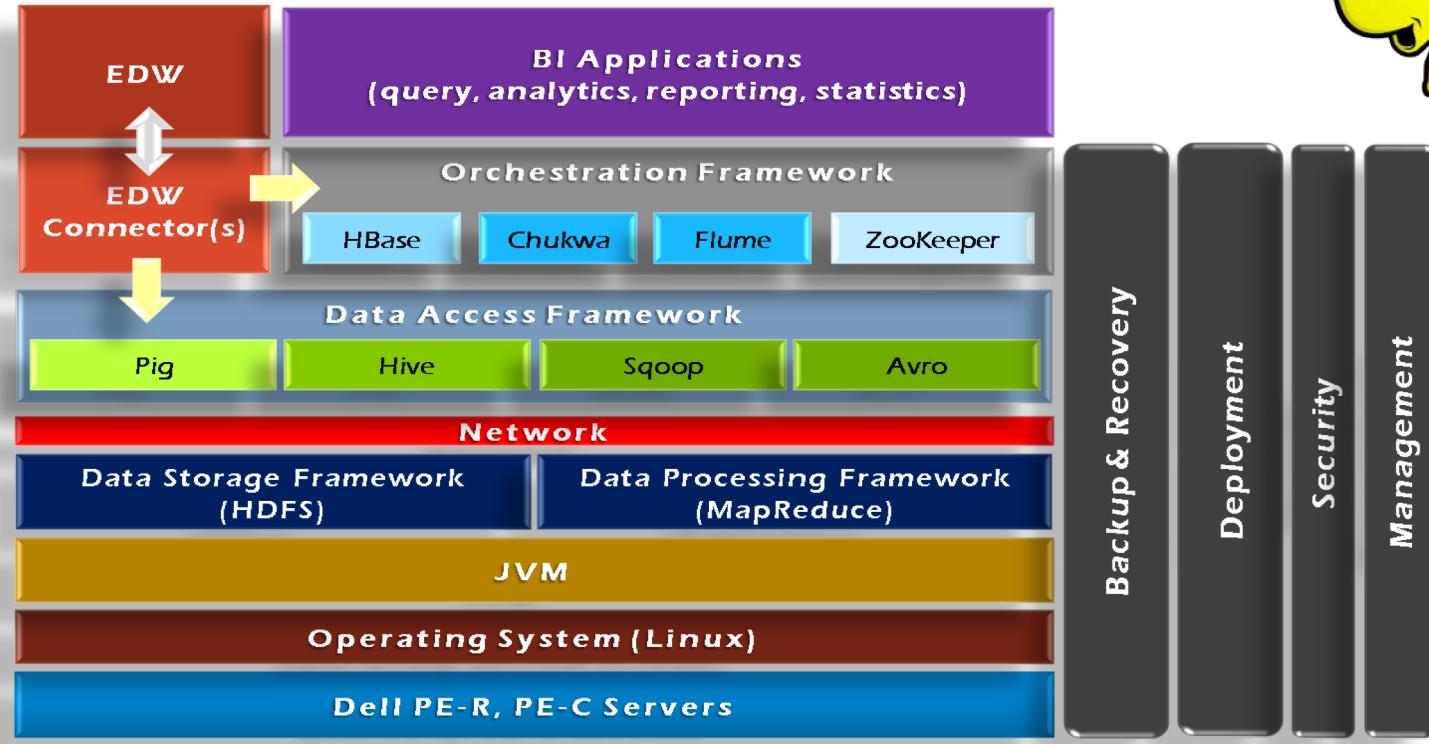
- An open-source software framework that supports data-intensive distributed applications, licensed under the Apache v2 license.

- **Goals / Requirements:**

- Abstract and facilitate the storage and processing of large and/or rapidly growing data sets
 - Structured and non-structured data
 - Simple programming models
- High scalability and availability
- Use commodity (cheap!) hardware with little redundancy
- Fault-tolerance
- Move computation rather than data

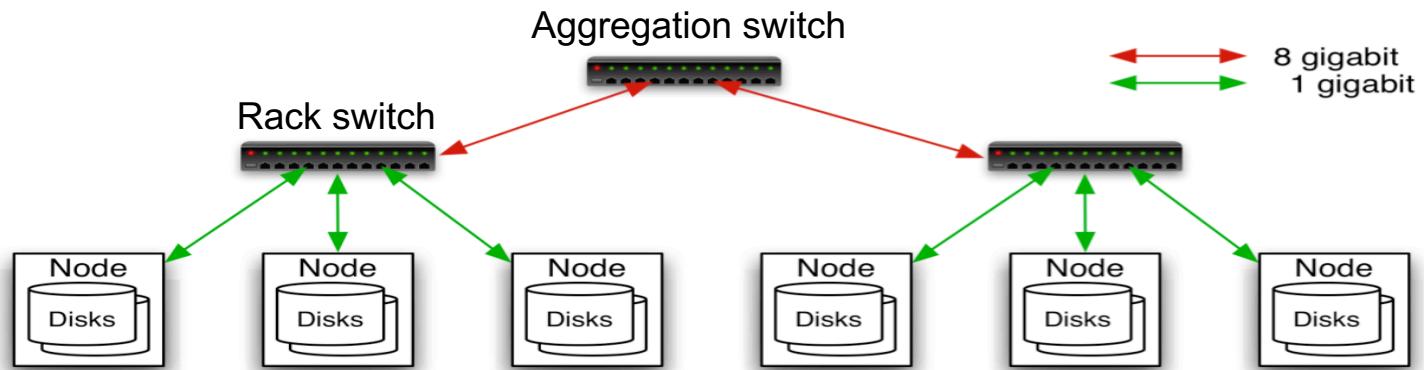
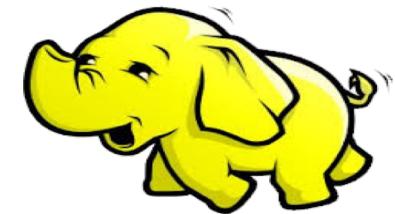


Hadoop Framework Tools

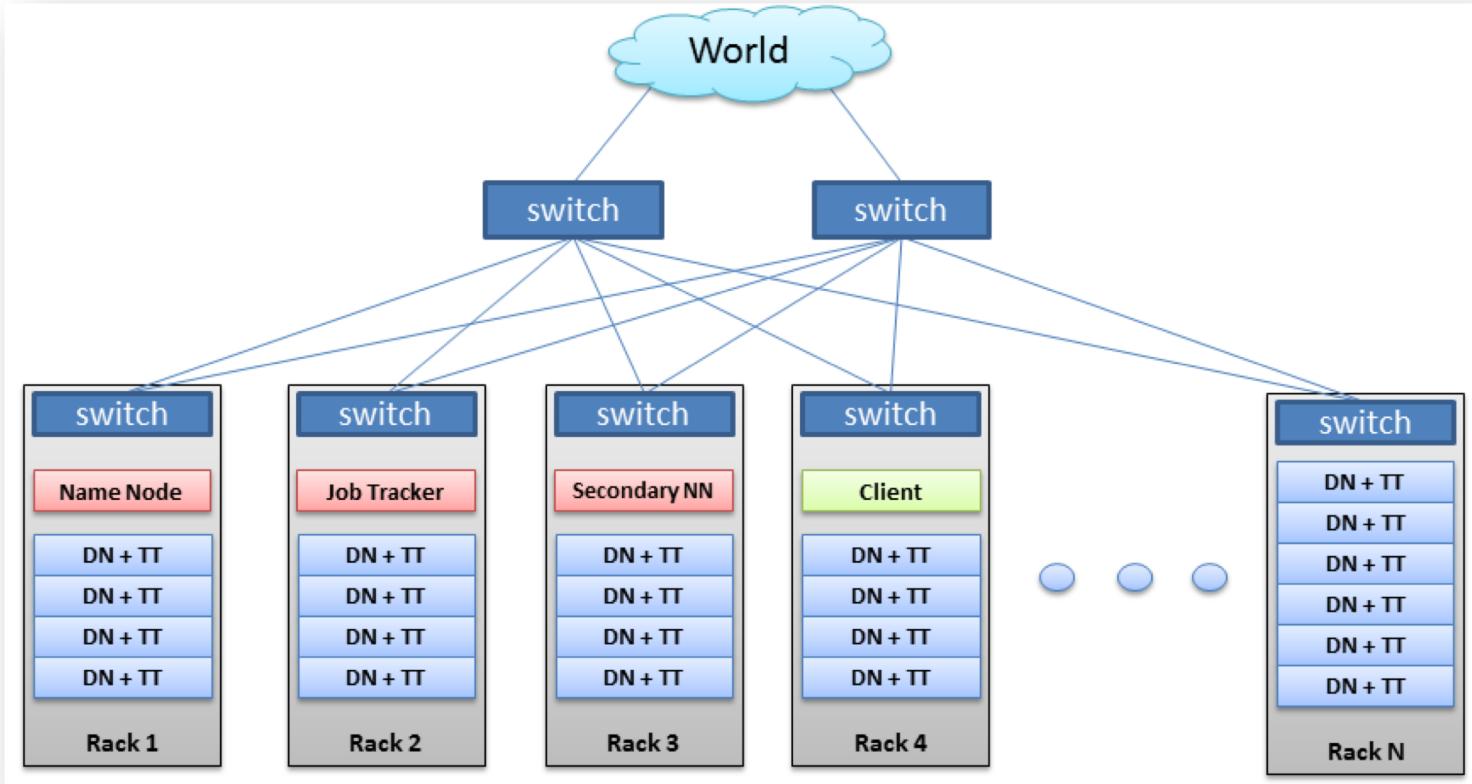


Cluster Hadoop

- Cluster creado con commodity Hardware:
 - Nodos inicialmente eran PCs
 - 30-40 nodos/rack
 - Red a 1 gigabit/s en rack



Arquitectura de almacenamiento HDFS

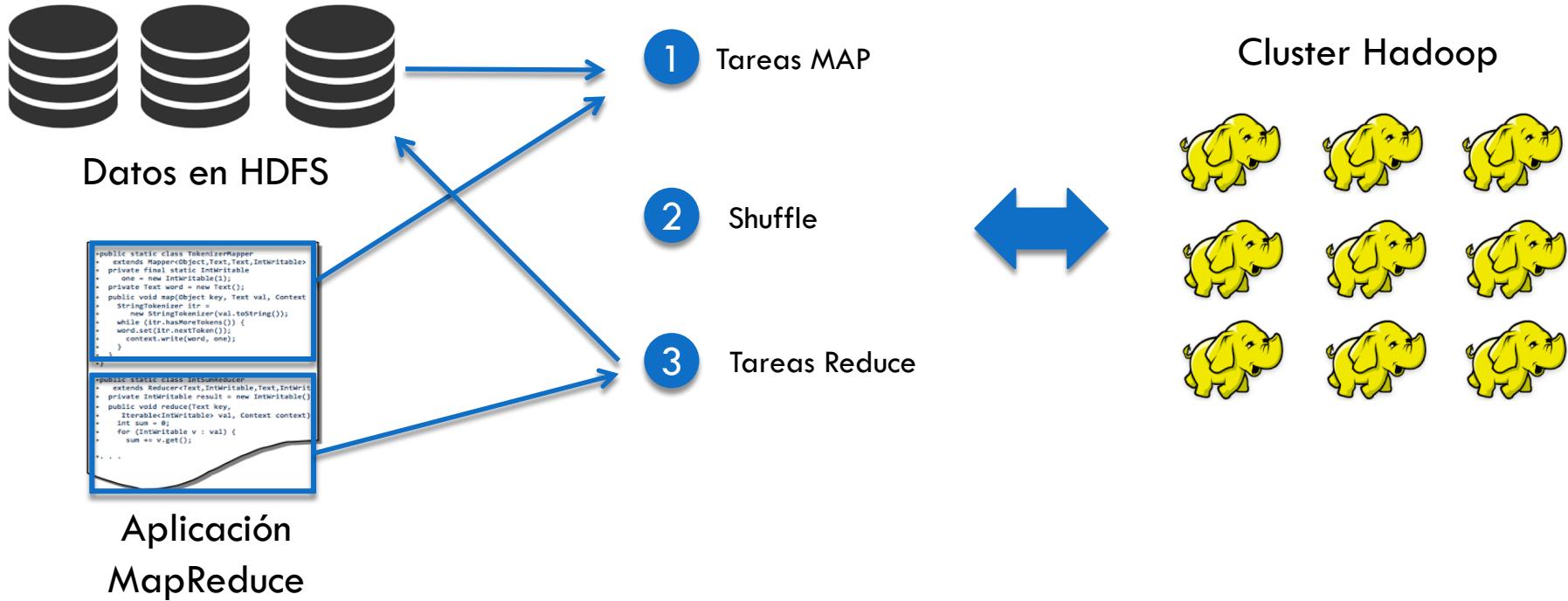


HDFS : Hadoop Distributed File System



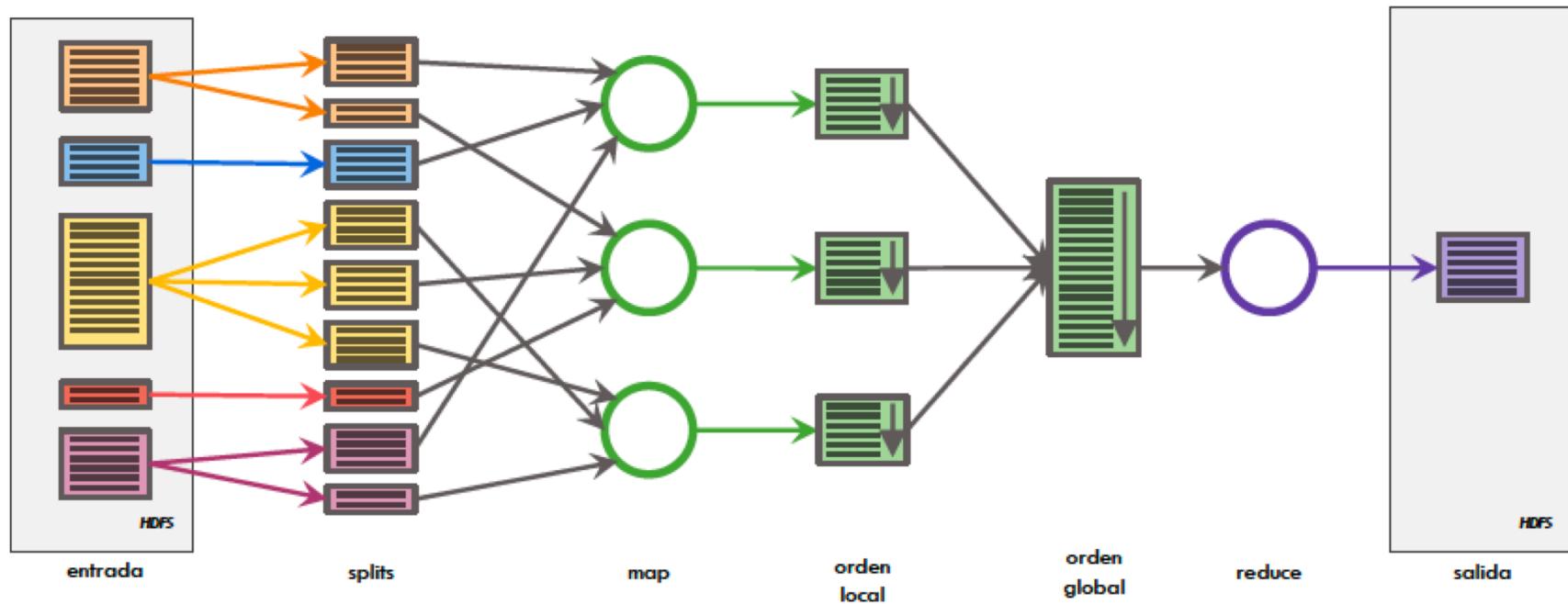
- Sistema de Ficheros distribuido muy grande
 - 10K nodos, 100 millones de ficheros 10PB
- Realizado con “*Commodity Hardware*”
 - Ficheros replicados para tolerancia a fallos
 - Detecta fallos y recupera los datos.
- Optimizado para proceso por lotes (“*Batch Processing*”).
 - Expone la localización de los datos y así permite que la computación se pueda llevar cerca de los datos.
 - El ancho de banda agregado es muy alto.

Cómo se ejecuta una aplicación Hadoop



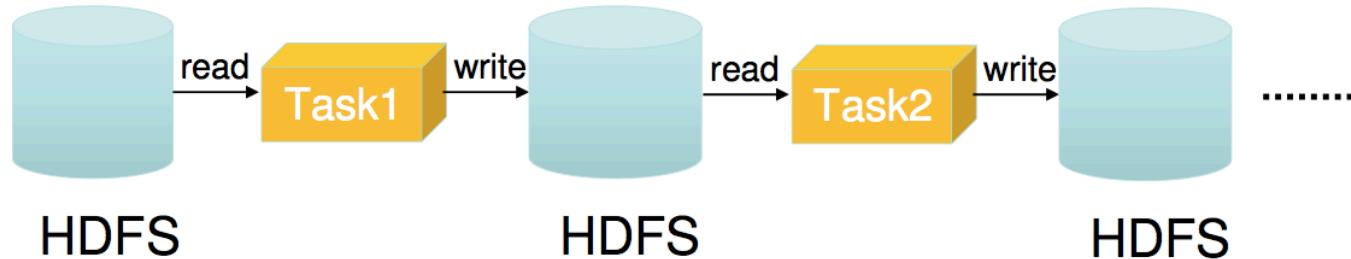
Cómo se ejecuta una aplicación Hadoop

- ¿Cómo se ejecutan las tareas?

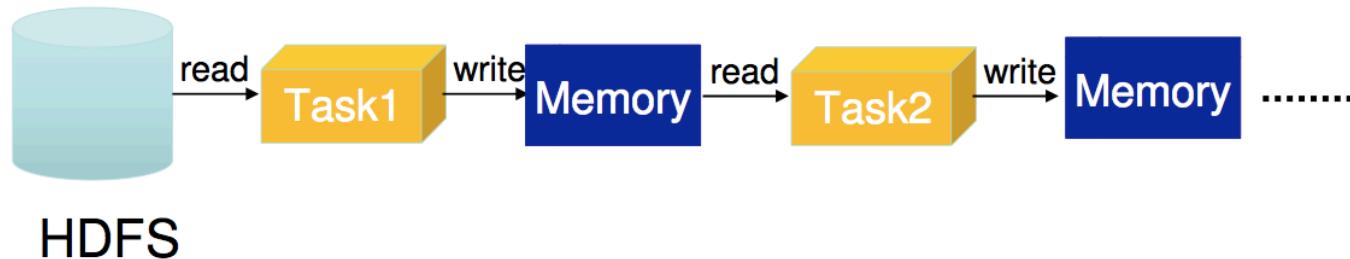


Intro: MapReduce vs Spark

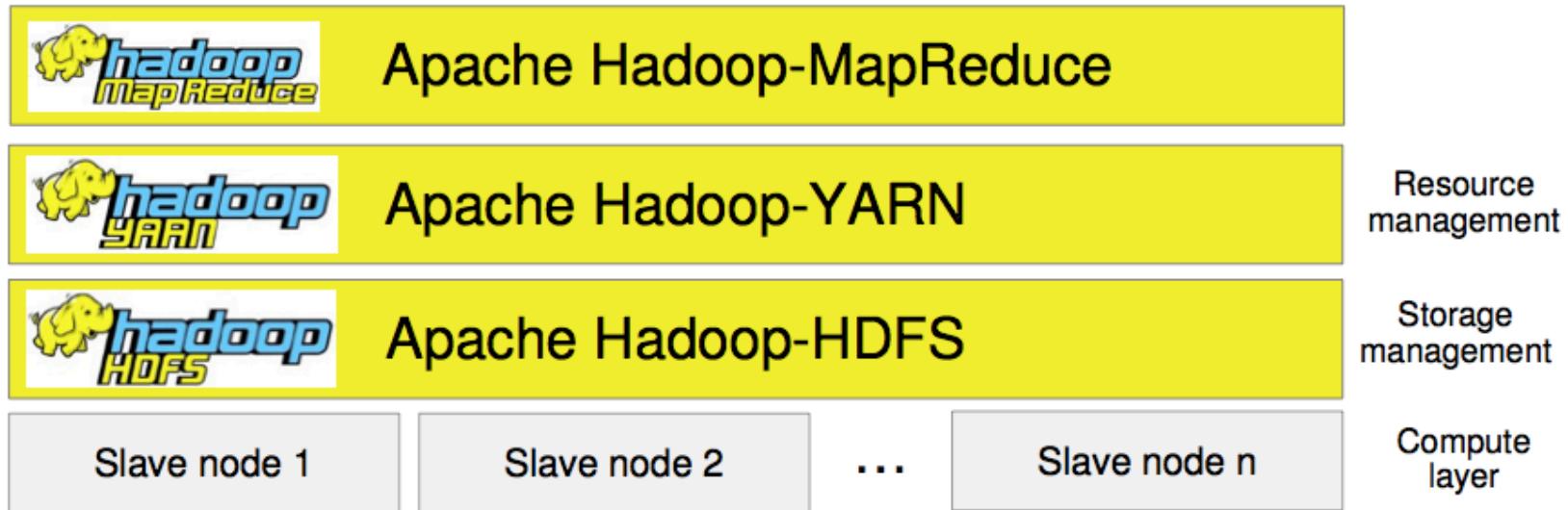
- Map/Reduce:



- Spark:



Intro: Arquitectura Hadoop



¿Qué necesitamos?

- Una ordenador con software para crear máquinas virtuales y conexión a Internet
 - La instalación en una máquina real es equivalente
 - Usaremos VMware
- Una ISO de Linux
 - Usaremos CentOS que es una RedHat “opensource”
- Hadoop
 - Este y otro software relacionado lo descargaremos una vez instalado Linux

Instalación de Hadoop

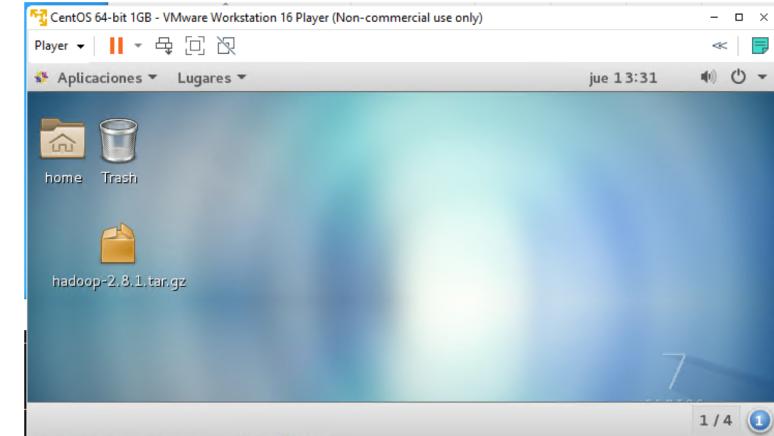
- Partimos de una imagen de MV que encontraréis en cada equipo
 - Recomendación: cambiar la configuración de la MV para “darle potencia”, ya que los equipos del laboratorio nos lo permiten
 - 2 GB de RAM
 - 2 procesadores
 - Acceso:
 - bigdata (bigdata)
 - root (bigdata)
-  Antes de empezar, copiaros la imagen de MV tal como la encontréis en vuestro equipo, pues la utilizaremos varias veces como punto de partida

Máquina virtual de partida

- Instalación “minimalista”
 - CentOS 7 (ISO disponible en los equipos del laboratorio)
 - 2 GB de RAM
 - 1 procesador
 - Instalación paquete de software “Compute node”
 - Se añade:
 - Herramientas de monitorización de HW
 - Herramientas de rendimiento
 - Administración remota de Linux
 - Herramientas de desarrollo
 - Habilitar interfaz de red

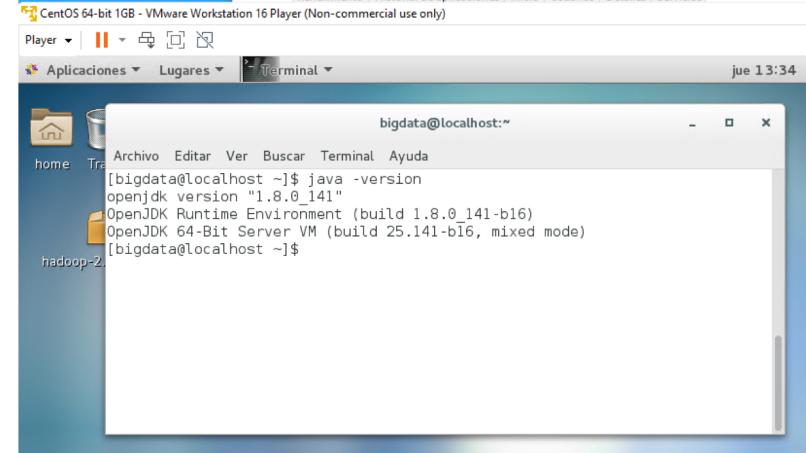
Máquina virtual de partida

- En esta máquina virtual ya se ha descargado en el escritorio:
 - `wget apache.rediris.es/hadoop/common/hadoop-2.8.1/hadoop-2.8.1.tar.gz`
- Actualmente versiones mas modernas
- Los requisitos de hadoop 2.8
 - Versión de java 1.7 comprobar con :
 - `java -versión`
 - `ls -la /usr/lib/jvm`
- Y si no coincide instalar:
`java-1.7.0-openjdk`



Máquina virtual de partida

- los requisitos de hadoop 2.8
 - Versión de java 1.7 comprobar con :
 - java -versión



```
CentOS 64-bit 1GB - VMware Workstation 16 Player (Non-commercial use only)
Player | ||| Terminal
Aplicaciones Lugares Terminal
bigdata@localhost:~$ Archivo Editar Ver Buscar Terminal Ayuda
[bigdata@localhost ~]$ java -version
openjdk version "1.8.0_141"
OpenJDK Runtime Environment (build 1.8.0_141-b16)
OpenJDK 64-Bit Server VM (build 25.141-b16, mixed mode)
[bigdata@localhost ~]$
```

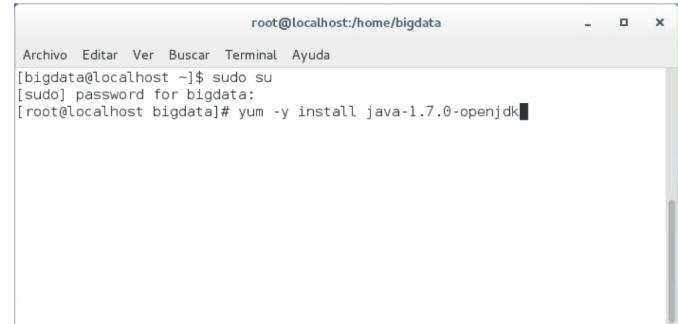
Y si no coincide instalar:

- sudo yum -y install java-1.7.0-openjdk

Se necesita ser root.

Instalar también:

- sudo yum -y install rsync



```
root@localhost:/home/bigdata
Archivo Editar Ver Buscar Terminal Ayuda
[bigdata@localhost ~]$ sudo su
[sudo] password for bigdata:
[root@localhost bigdata]# yum -y install java-1.7.0-openjdk
```

Instalar Hadoop

- La 1^a instalación es Single Node (una sola máquina)
- Recordamos los requisitos ya instalados:
 - Java 7. La opción más sencilla en OpenJDK que viene con CentOS
 - > yum -y install java-1.7.0-openjdk
 - SSH. Viene con Linux, pero necesitamos rsync
 - > yum -y install rsync
- Y también se obtuvo hadoop con:
 - > wget apache.rediris.es/hadoop/common/hadoop-2.8.1/hadoop-2.8.1.tar.gz

Otras versiones ver <https://apache.rediris.es/hadoop/common>

Instalar Hadoop

- Descomprimir hadoop
 - > cd /home/bigdata/Escritorio
 - > tar xvzf hadoop-2.8.1.tar.gz
- Movemos a /opt
 - > sudo mv hadoop-2.8.1 /opt
- Creamos un link a la carpeta de hadoop
 - Permite elegir entre diferentes versiones
 - > cd /opt
 - > sudo ln -s hadoop-2.8.1 hadoop
 - > cd hadoop
- Editar fichero etc/hadoop/hadoop-env.sh
 - export JAVA_HOME=/usr/lib/jvm/jre-1.7.0-openjdk

The image shows two terminal windows side-by-side. The left window is titled 'root@localhost:/home/bigdata/Escritorio' and shows the command history for extracting the Hadoop tar file. The right window is titled 'root@localhost:/opt' and shows the command history for moving the Hadoop directory and creating a symbolic link.

```
root@localhost:/home/bigdata/Escritorio
Archivo Editar Ver Buscar Terminal Ayuda
[root@localhost bigdata]# ls
cloudera-manager-installer.bin Documentos Imágenes Plantillas Vídeos
Descargas Escritorio Música Público
[root@localhost bigdata]# cd /home/bigdata/Escritorio/
[root@localhost Escritorio]# ls
hadoop-2.8.1.tar.gz
[root@localhost Escritorio]# tar xvzf hadoop-2.8.1.tar.gz

root@localhost:/opt
Archivo Editar Ver Buscar Terminal Ayuda
hadoop-2.8.1/include/StringUtil.hh
hadoop-2.8.1/include/SerialUtil.hh
hadoop-2.8.1/LICENSE.txt
hadoop-2.8.1/NOTICE.txt
hadoop-2.8.1/README.txt
[root@localhost Escritorio]# ls
hadoop-2.8.1 hadoop-2.8.1.tar.gz
[root@localhost Escritorio]# mv hadoop-2.8.1 /opt/
[root@localhost Escritorio]# cd /opt/
[root@localhost opt]# ls
hadoop-2.8.1 rh
[root@localhost opt]# ln -s hadoop-2.8.1/ hadoop
[root@localhost opt]# ls -l
total 0
lrwxrwxrwx 1 root root 13 ago 5 14:01 hadoop -> hadoop-2.8.1/
drwxrwxr-x 9 500 500 149 jun 2 2017 hadoop-2.8.1
drwxr-xr-x. 2 root root 6 mar 26 2015 rh
[root@localhost opt]#
```

Instalar Hadoop

- Editar fichero `etc/hadoop/hadoop-env.sh`
 - `export JAVA_HOME=/usr/lib/jvm/jre-1.7.0-openjdk`

```
root@localhost:/opt/hadoop
Archivo Editar Ver Buscar Terminal Ayuda
hadoop-2.8.1/README.txt
[root@localhost Escritorio]# ls
hadoop-2.8.1  hadoop-2.8.1.tar.gz
[root@localhost Escritorio]# mv hadoop-2.8.1 /opt/
[root@localhost Escritorio]# cd /opt/
[root@localhost opt]# ls
hadoop-2.8.1  rh
[root@localhost opt]# ln -s hadoop-2.8.1/ hadoop
[root@localhost opt]# ls -l
total 0
lrwxrwxrwx 1 root root 13 ago  5 14:01 hadoop -> hadoop-2.8.1/
drwxrwxr-x  9 500 500 149 jun  2 2017 hadoop-2.8.1
drwxr-xr-x. 2 root root   6 mar 26 2015 rh
[root@localhost opt]# cd hadoop
[root@localhost hadoop]# ls
bin  include  libexec  NOTICE.txt  sbin
etc  lib     LICENSE.txt  README.txt  share
[root@localhost hadoop]# vim etc/hadoop/hadoop-env.sh
```

```
root@localhost:/opt/hadoop
Archivo Editar Ver Buscar Terminal Ayuda
# optional. When running a distributed configuration it is best to
# set JAVA_HOME in this file, so that it is correctly defined on
# remote nodes.

# The java implementation to use.
#export JAVA_HOME=${JAVA_HOME}
export JAVA_HOME=/usr/lib/jvm/jre-1.7.0-openjdk
# The jsvc implementation to use. Jsvc is required to run secure datanodes
# that bind to privileged ports to provide authentication of data transfer
# protocol. Jsvc is not required if SASL is configured for authentication o
f
# data transfer protocol using non-privileged ports.
#export JSVC_HOME=${JSVC_HOME}

export HADOOP_CONF_DIR=${HADOOP_CONF_DIR:-"/etc/hadoop"}

# Extra Java CLASSPATH elements. Automatically insert capacity-scheduler.
-- INSERTAR --
25,1          20%
```

Instalar Hadoop

- Hemos descomprimido el código de hadoop dentro de nuestro sistema de ficheros local
- ¿Por qué no hay que compilar nada?
 - Hadoop es código JAVA!
- Conviene chequear compatibilidad con tu versión de Java
 - <http://hadoop.apache.org/releases.html>
 - <http://hadoop.apache.org/release>
 - <https://wiki.apache.org/hadoop/HadoopJavaVersions>

Instalar Hadoop

- <http://hadoop.apache.org/release>
- <https://wiki.apache.org/hadoop/HadoopJavaVersions>

← → ⌂ https://cwiki.apache.org/confluence/display/HADOOP2/HadoopJavaVersions

Confluence Espacios Buscar

- HadoopDfsReadWriteExample
- HadoopHackathon20110518
- Hadoop in China Salon
- Hadoop in China Summit
- HadoopIPv6
- HadoopIsNot
- **HadoopJavaVersions**
- HadoopMapRedClasses
- HadoopMapReduce
- HadoopMapReduceConfigureMet
- HadoopMapReduceSecondaryDat
- HadoopMapReduceSequenceFilef
- HadoopMapReduceTips
- HadoopOverview

Panel / Home

HadoopJavaVersions

Creado por ASF Infrabot el jul 09, 2019

Moved to Confluence Wiki: <https://cwiki.apache.org/confluence/display/HADOOP/Hadoop+Java+Versions>

The following contents are deprecated.

Hadoop Java Versions

Version 2.7 and later of Apache Hadoop [requires Java 7](#). It is built and tested on both OpenJDK and Oracle (HotSpot)'s JDK/JRE.

Earlier versions (2.6 and earlier) support Java 6.

Tested JDK

Here are the known JDKs in use or which have been tested:

Version	Status	Reported By
---------	--------	-------------

El lenguaje de programación Java

➤ Máquina Virtual de Java (JVM)

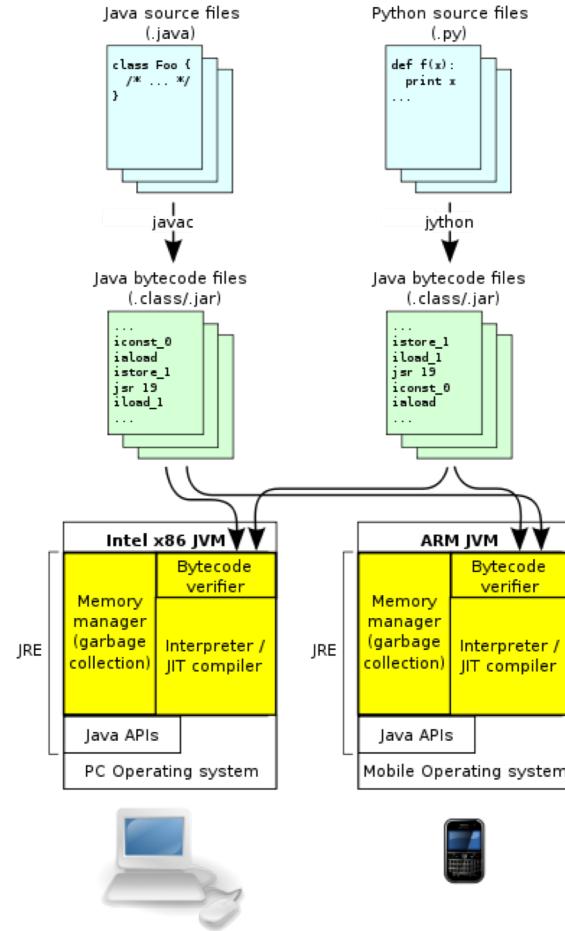
➤ Portabilidad

- Cualquier dispositivo
- Cualquier SO
- Java > ByteCode > Código ejecutable

➤ Menor rendimiento

- Etapas intermedias de ejecución
- Dificultad de programación MP/MC

➤ Aislamiento

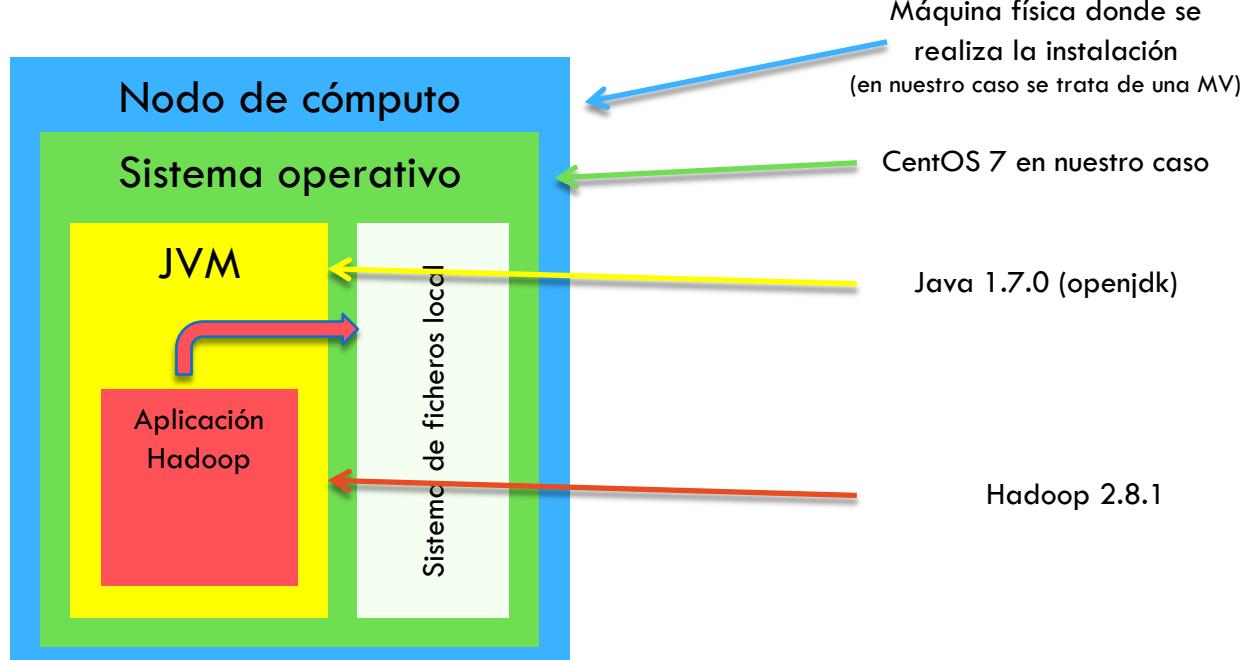


Instalación de Hadoop

- Instalación en modo *Standalone*
 - No se ejecutan demonios
 - Todo se ejecuta en una única Máquina Virtual de Java (MVJ)
 - No se usa HDFS
 - Adecuado para desarrollo y debug de aplicaciones

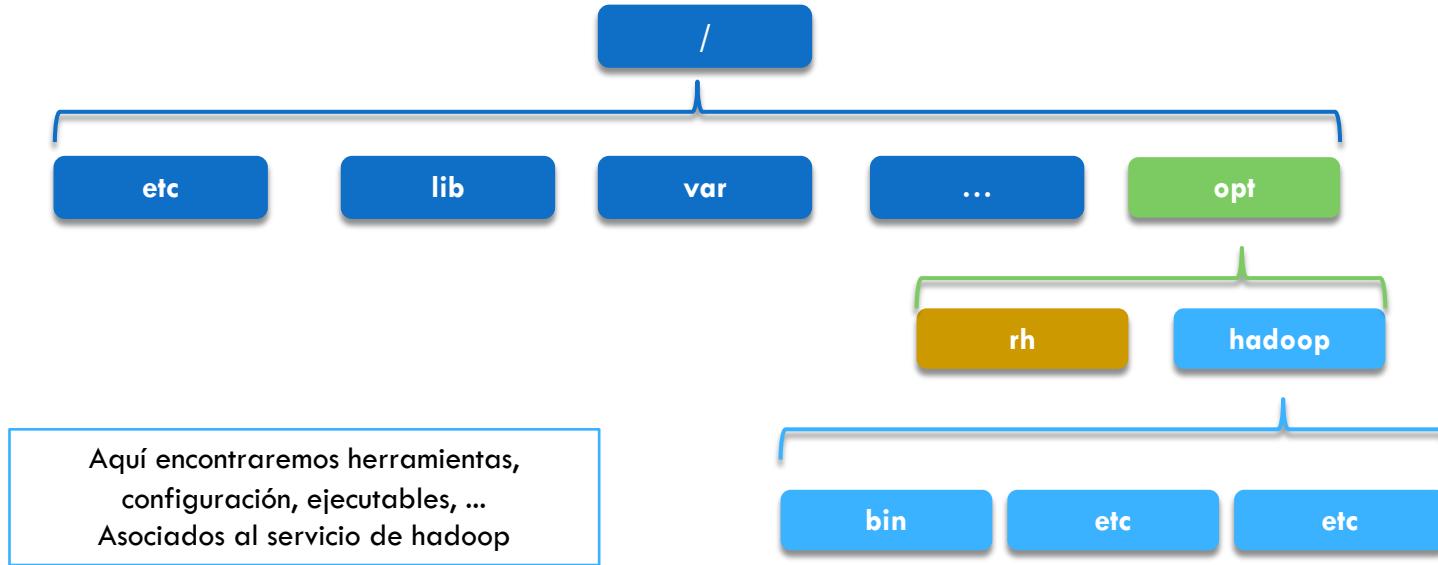
Instalación de Hadoop

➤ Instalación en modo *Standalone*



Instalación de Hadoop

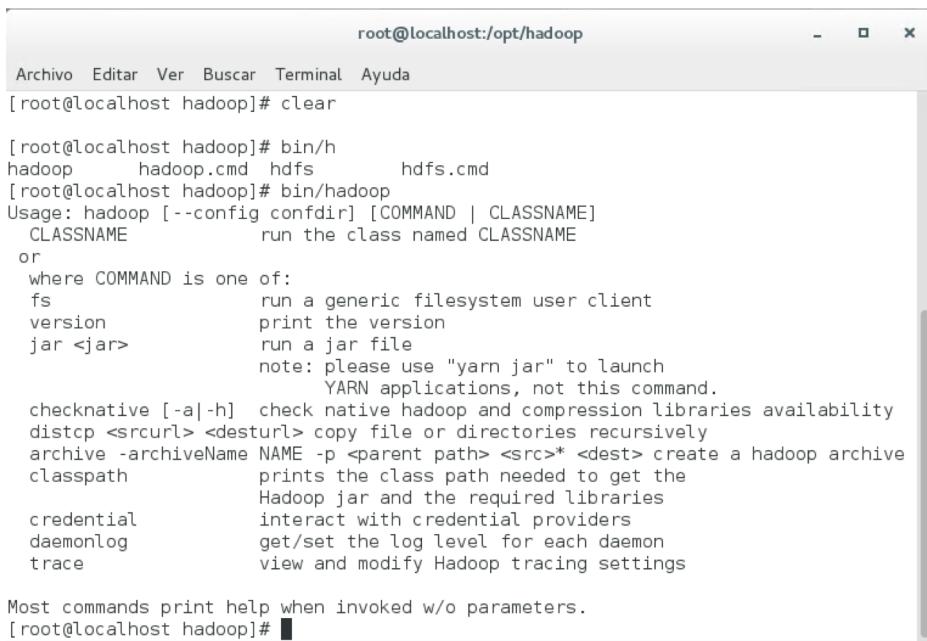
➤ Instalación en modo *Standalone*



Instalar Hadoop - Standalone

- Por defecto Hadoop está configurado para ejecutarse en modo non-distributed, como un único proceso java
- Ejemplo 1

- > cd /opt/hadoop
- > bin/hadoop



The screenshot shows a terminal window titled "root@localhost:/opt/hadoop". The window has a standard Linux terminal interface with tabs for Archivo, Editar, Ver, Buscar, Terminal, and Ayuda. The command-line history shows the user navigating to the Hadoop directory and running the "hadoop" command. The output of the "hadoop" command is displayed, providing usage information for various Hadoop subcommands like fs, version, jar, and checknative.

```
root@localhost:/opt/hadoop
Archivo Editar Ver Buscar Terminal Ayuda
[root@localhost hadoop]# clear
[root@localhost hadoop]# bin/
hadoop      hadoop.cmd  hdfs      hdfs.cmd
[root@localhost hadoop]# bin/hadoop
Usage: hadoop [--config confdir] [COMMAND | CLASSNAME]
CLASSNAME          run the class named CLASSNAME
or
where COMMAND is one of:
fs                  run a generic filesystem user client
version            print the version
jar <jar>           run a jar file
note: please use "yarn jar" to launch
      YARN applications, not this command.
checknative [-a|-h]  check native hadoop and compression libraries availability
distcp <srcurl> <desturl> copy file or directories recursively
archive -archiveName NAME -p <parent path> <src>* <dest> create a hadoop archive
classpath          prints the class path needed to get the
                  Hadoop jar and the required libraries
credential         interact with credential providers
daemonlog          get/set the log level for each daemon
trace              view and modify Hadoop tracing settings

Most commands print help when invoked w/o parameters.
[root@localhost hadoop]#
```

Instalar Hadoop - Standalone

➤ Ejemplo 1

➤ > bin/hadoop

jar

Clase a invocar

share/hadoop/tools/lib/hadoop-streaming-2.8.1.jar

-input prueba

Ficheros (en este caso directorio)

-output salida

Directorio de salida (no debe existir)

-mapper cat

Qué hacer en las fase map y reduce

-reducer wc

Parámetros de la clase

Instalar Hadoop - Standalone

- Ejemplo1: Prepara un directorio con los datos(para contar líneas | palabras | caracteres)

- > mkdir prueba
- > cp etc/hadoop/*.xml prueba

No se
usa
HDFS!!

Ejecuta el comando

- > bin/hadoop jar share/hadoop/tools/lib/hadoop-streaming-2.8.1.jar -input prueba -output salida -mapper cat -reducer wc

```
root@localhost:/opt/hadoop
Archivo Editar Ver Buscar Terminal Ayuda
Streaming Command Failed!
[root@localhost hadoop]# clear

[root@localhost hadoop]# ls prueba/
capacity-scheduler.xml  hadoop-policy.xml  httpfs-site.xml  kms-site.xml
core-site.xml            hdfs-site.xml    kms-acls.xml   yarn-site.xml
[root@localhost hadoop]# bin/hadoop jar share/hadoop/tools/lib/hadoop-streaming-2.8.1
.jar -input prueba -output salida -mapper cat -reducer wc
21/08/05 14:30:49 INFO Configuration.deprecation: session.id is deprecated. Instead,
use dfs.metrics.session-id
21/08/05 14:30:49 INFO jvm.JvmMetrics: Initializing JVM Metrics with processName=JobT
racker, sessionId=
21/08/05 14:30:49 INFO jvm.JvmMetrics: Cannot initialize JVM Metrics with processName
=JobTracker, sessionId= - already initialized
21/08/05 14:30:49 INFO mapred.FileInputFormat: Total input files to process : 8
21/08/05 14:30:49 INFO mapreduce.JobSubmitter: number of splits:8
21/08/05 14:30:49 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local13
26438792_0001
21/08/05 14:30:49 INFO mapreduce.Job: The url to track the job: http://localhost:8080
/
```

Instalar Hadoop - Standalone

➤ Ejemplo 1

➤ Comprobar la salida

➤ > cat salida/part-00000

➤ Comparar resultado con el siguiente comando

➤ > wc prueba/*

```
Bytes Written=3/
21/08/05 14:30:52 INFO streaming.StreamJob: Output directory: salida
[root@localhost hadoop]#
[root@localhost hadoop]#
[root@localhost hadoop]#
[root@localhost hadoop]# ls
bin include libexec NOTICE.txt README.txt sbin
etc lib LICENSE.txt prueba salida share
[root@localhost hadoop]# ls -l salida/
total 4
-rw-r--r-- 1 root root 25 ago  5 14:30 part-00000
-rw-r--r-- 1 root root  0 ago  5 14:30 _SUCCESS
[root@localhost hadoop]# cat salida/part-00000
    757    2803   27305
[root@localhost hadoop]# cat prueba/* |wc
    757    2803   26548
[root@localhost hadoop]#
```

Instalar Hadoop - Standalone

- Ejemplo 1 - Observaciones
 - NO hemos utilizado HDFS en ningún momento
 - Sólo se ha utilizado el FS local
 - Directorios “prueba” y “salida”
 - Repetir la ejecución de la prueba mientras en otra terminal ejecutas el comando “top”

```
done. And is in the process of committing
.7/09/13 04:31:50 INFO mapred.LocalJobRunner: Records R/W=173/1
.7/09/13 04:31:50 INFO mapred.Task: Task 'attempt_local1212135577_0001_m_000001_0'
done.
.7/09/13 04:31:50 INFO mapred.LocalJobRunner: Finishing task: attempt_local12121355
7_0001_m_000001_0
.7/09/13 04:31:50 INFO mapred.LocalJobRunner: Starting task: attempt_local121213557
_0001_m_000002_0
.7/09/13 04:31:50 INFO output.FileOutputCommitter: File Output Committer Algorithm
version is 1
.7/09/13 04:31:50 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanupUp
Temporary folders under output directory:false, ignore cleanup failures: false
.7/09/13 04:31:50 INFO mapred.Task: Using ResourceCalculatorProcessTree : []
.7/09/13 04:31:50 INFO mapred.MapTask: Processing split: file:/opt/hadoop-2.8.1/pru
ba/capacity-scheduler.xml:0:4942
.7/09/13 04:31:50 INFO mapred.MapTask: numReduceTasks: 1
.7/09/13 04:31:50 INFO mapreduce.Job: map 100% reduce 0%
.7/09/13 04:31:50 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
.7/09/13 04:31:50 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
.7/09/13 04:31:50 INFO mapred.MapTask: soft limit at 83886080
.7/09/13 04:31:50 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
.7/09/13 04:31:50 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
.7/09/13 04:31:50 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.lib.WritableCompositeValueCollector
```

```
top - 04:31:48 up 9:29, 3 users, load average: 0,74, 0,27, 0,17
Tasks: 177 total, 1 running, 176 sleeping, 0 stopped, 0 zombie
%Cpu(s): 85,6 us, 14,0 sy, 0,0 ni, 0,0 id, 0,3 wa, 0,0 hi, 0,0 si, 0,0 st
KiB Mem : 999920 total, 79424 free, 661000 used, 259496 buff/cache
KiB Swap: 2097148 total, 2037544 free, 59604 used, 118044 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S %CPU	%MEM	TIME+ COMMAND
30084	bigdata	20	0	2244584	136928	20390	S 61,8	13,7	0:04.27 java
12124	bigdata	20	0	1492928	174780	12356	S 17,3	17,5	2:26.88 gnome-shell
1134	root	20	0	250144	42440	2324	S 8,3	4,2	1:03.66 Xorg
13670	bigdata	20	0	565324	14952	5208	S 1,7	1,5	0:21.34 gnome-terminal
25	root	20	0	0	0	0	S 0,7	0,0	0:10.65 kswapd0
709	root	20	0	302772	1836	1308	S 0,3	0,2	0:43.74 vmtoolsd
11883	bigdata	20	0	35988	1876	560	S 0,3	0,2	0:00.90 dbus-daemon
12204	bigdata	20	0	575276	5848	1564	S 0,3	0,6	0:04.83 caribou
29468	root	20	0	157704	2300	1568	R 0,3	0,2	0:03.77 top
1	root	20	0	128088	4252	2452	S 0,0	0,4	0:09.65 systemd
2	root	20	0	0	0	0	S 0,0	0,0	0:01.79 kthread
3	root	20	0	0	0	0	S 0,0	0,0	0:01.81 ksoftirqd/0
7	root	rt	0	0	0	0	S 0,0	0,0	0:00.00 migration/0
8	root	20	0	0	0	0	S 0,0	0,0	0:00.00 rcu_bh
9	root	20	0	0	0	0	S 0,0	0,0	0:03.30 rcu_sched
10	root	rt	0	0	0	0	S 0,0	0,0	0:16.53 watchdog/0

Instalar Hadoop - Standalone

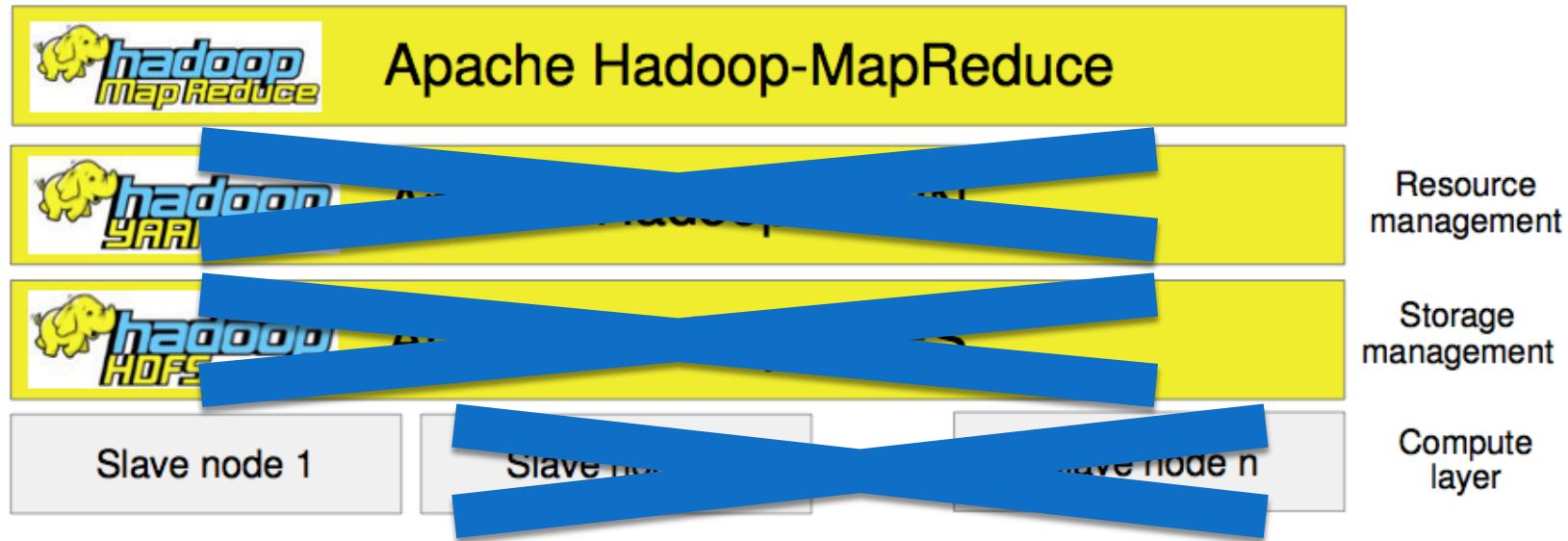
➤ Ejemplo2

- Ejecutamos un wordcount sobre ficheros de texto (por ejemplo los del directorio de prueba) pero con otra clase jar...
 - Pista utilizar:
 - share/hadoop/mapreduce/hadoop-mapreduce-examples-2.8.1.jar

SOL: bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.8.1.jar wordcount prueba salida1

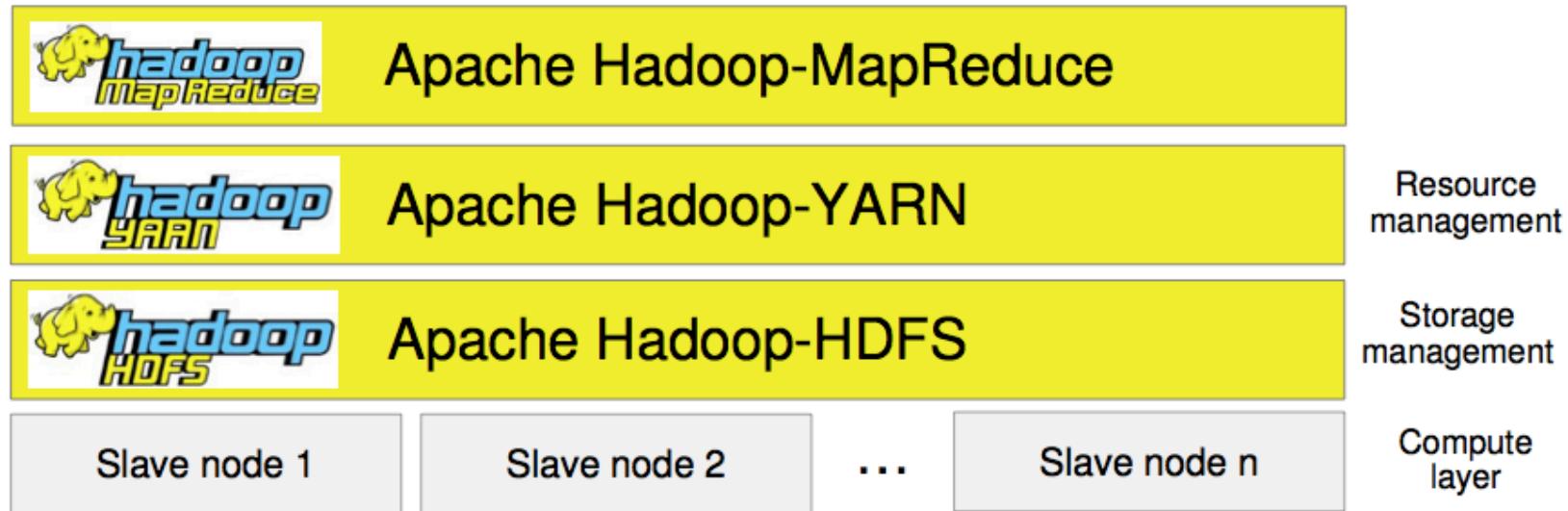
Instalación de Hadoop

- Modo *standalone instalado*



Instalación de Hadoop

- Este no es el modo de ejecución de Hadoop del que estamos acostumbrados a hablar



Instalación de Hadoop

- Hadoop se puede instalar de tres maneras distintas:
 - Standalone
 - No se ejecutan demonios
 - Todo se ejecuta en una única Máquina Virtual de Java (MVJ)
 - No se usa HDFS
 - Adecuado para desarrollo y debug de aplicaciones

Instalación de Hadoop

- Hadoop se puede instalar de tres maneras distintas:
 - Pseudo-Distributed
 - Todos los demonios se ejecutan en la misma máquina
 - En su propia MVJ
 - Se emplea HDFS
 - Adecuado para simular un cluster en una sola máquina y para debug de programas antes de llevarlos a un “cluster real”

Instalar Hadoop – Pseudo-Distributed

- Antes de nada debemos configurar ssh para funcionar sin contraseña para conexiones a la misma máquina (localhost)
 - En esta configuración los diferentes procesos de Hadoop hacen uso de la red para conectarse aunque estén en la misma máquina
 - > ssh-keygen (pulso Intro a todo)
 - > ssh-copy-id localhost (esto vale para cualquier máquina)
- Comprobar que se puede conectar
 - > ssh localhost

The image shows two terminal windows. The top window is titled 'bigdata@localhost:/home/bigdata' and displays the output of the 'ssh-keygen' command. It shows the generation of an RSA key pair, saving it to '/root/.ssh/id_rsa', and printing the public key to '/root/.ssh/id_rsa.pub'. The bottom window is titled 'root@localhost:' and shows the output of the 'ssh-copy-id' command. It attempts to copy the user's public SSH key to the remote host 'localhost' and prompts for confirmation. Finally, it logs in successfully with the message 'Last login: Thu Aug 5 16:52:09 2021'.

```
bigdata@localhost:/home/bigdata
Archivo Editar Ver Buscar Terminal Ayuda
[root@localhost bigdata]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
24:ba:bb:8e:86:d2:c9:dc:9e:58:d2:7c:65:71:c6 root@localhost.localdomain
The key's randomart image is:
+--[ RSA 2048]----+
|          . |
|         . o E |
|        . o + |
|       . S |
|      o . o |
|     +...= . |
|    o =*= |
|   ...oX. |
+-----+
[root@localhost bigdata]#
```



```
root@localhost:~
Archivo Editar Ver Buscar Terminal Ayuda
|o =*.= | |
|..o=X. | |
+-----+
[root@localhost bigdata]# ssh-copy-id localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is a0:d9:0f:06:4e:2b:0f:b7:31:44:c7:6d:0b:6c:53:1c.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out
any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted
now it is to install the new keys
root@localhost's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'localhost'"
and check to make sure that only the key(s) you wanted were added.

[root@localhost bigdata]# ssh localhost
Last login: Thu Aug 5 16:52:09 2021
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
```

Instalar Hadoop – Pseudo-Distributed

- Añade la siguiente configuración a `etc/hadoop/core-site.xml`:

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://localhost:9000</value>
  </property>
</configuration>
```

Importante: Se parte del directorio donde esta instalado Hadoop

A terminal window titled "bigdata@localhost:/opt/hadoop". The command history shows:

```
Archivo Editar Ver Buscar Terminal Ayuda
bigdata@localhost:/opt/hadoop - □ ×
[root@localhost opt]# pwd
/opt
[root@localhost opt]# ls
hadoop hadoop-2.8.1 rh
[root@localhost opt]# cd hadoop
[root@localhost hadoop]# pwd
/opt/hadoop
[root@localhost hadoop]# vim etc/hadoop/co
nfiguration.xsl container-executor.cfg core-site.xml
[root@localhost hadoop]# vim etc/hadoop/co
nfiguration.xsl container-executor.cfg core-site.xml
[root@localhost hadoop]# vim etc/hadoop/core-site.xml
```

A terminal window titled "bigdata@localhost:/opt/hadoop". The command history shows:

```
Archivo Editar Ver Buscar Terminal Ayuda
bigdata@localhost:/opt/hadoop - □ ×
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.
-->
<!-- Put site-specific property overrides in this file. -->
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://localhost:9000</value>
  </property>
</configuration>
-- INSERTAR --
```

The status bar at the bottom right shows "24,16 Final".

Instalar Hadoop – Pseudo-Distributed

- Añade la siguiente configuración a `etc/hadoop/hdfs-site.xml`:

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
</configuration>
```

Importante: Se parte del directorio donde esta instalado Hadoop

```
bigdata@localhost:/opt/hadoop
Archivo Editar Ver Buscar Terminal Ayuda
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.
-->
<!-- Put site-specific property overrides in this file. -->
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://localhost:9000</value>
  </property>
</configuration>
[root@localhost hadoop]# vim etc/hadoop/hdfs-site.xml
```

```
bigdata@localhost:/opt/hadoop
Archivo Editar Ver Buscar Terminal Ayuda
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.
-->
<!-- Put site-specific property overrides in this file. -->
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
</configuration>
~
-- INSERTAR --
```

23,12-19 Final

Instalar Hadoop – Pseudo-Distributed

- Formatear sistema de ficheros
 - > bin/hdfs namenode -format
- Iniciar el NameNode y DataNode
 - > sbin/start-dfs.sh
(como root)

```
bigdata@localhost:/opt/hadoop
Archivo Editar Ver Buscar Terminal Ayuda
21/08/05 17:21:36 INFO util.GSet: Computing capacity for map BlocksMap
21/08/05 17:21:36 INFO util.GSet: VM type      = 64-bit
21/08/05 17:21:36 INFO util.GSet: 2.0% max memory 889 MB = 17.8 MB
21/08/05 17:21:36 INFO util.GSet: capacity      = 2^21 = 2097152 entries
21/08/05 17:21:36 INFO blockmanagement.BlockManager: dfs.block.access.token.enable=false
21/08/05 17:21:36 INFO blockmanagement.BlockManager: defaultReplication      = 1
21/08/05 17:21:36 INFO blockmanagement.BlockManager: maxReplication      = 512
21/08/05 17:21:36 INFO blockmanagement.BlockManager: minReplication      = 1
21/08/05 17:21:36 INFO blockmanagement.BlockManager: maxReplicationStreams = 2
21/08/05 17:21:36 INFO blockmanagement.BlockManager: replicationRecheckInterval = 3000
21/08/05 17:21:36 INFO blockmanagement.BlockManager: encryptDataTransfer      = false
21/08/05 17:21:36 INFO blockmanagement.BlockManager: maxNumBlocksToLog      = 1000
21/08/05 17:21:36 INFO namenode.FSNamesystem: fsOwner        = root (auth:SIMPLE)
21/08/05 17:21:36 INFO namenode.FSNamesystem: supergroup     = supergroup
21/08/05 17:21:36 INFO namenode.FSNamesystem: isPermissionEnabled = true
21/08/05 17:21:36 INFO namenode.FSNamesystem: HA Enabled: false
```

```
bigdata@localhost:/opt/hadoop
Archivo Editar Ver Buscar Terminal Ayuda
top - 17:27:18 up 37 min,  3 users,  load average: 0,05, 0,12, 0,13
Tasks: 182 total,   1 running, 181 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0,2 us,  0,3 sy,  0,0 ni, 99,5 id,  0,0 wa,  0,0 hi,  0,0 si,  0,0 st
KiB Mem : 1867276 total, 289060 free, 869416 used, 708800 buff/cache
KiB Swap: 2097148 total, 2097148 free,          0 used. 789396 avail Mem

PID USER    PR NI  VIRT  RES  SHR S %CPU %MEM TIME+ COMMAND
4157 bigdata  20  0 1969656 248640 56040 S  1,0 13,3  0:37,65 Web Content
692 root     20  0 382772 6256 4840 S  0,7  0,3  0:34,03 vmtoold
4069 bigdata  20  0 2220848 235708 67064 S  0,7 12,6  0:25,98 firefox
2929 bigdata  20  0 1672512 157376 47096 S  0,3  8,4  0:37,17 gnome-shell
  1 root     20  0 125304  3872 2416 S  0,0  0,2  0:00,91 systemd
  2 root     20  0      0  0    0 S  0,0  0,0  0:00,00 kthreadd
  3 root     20  0      0  0    0 S  0,0  0,0  0:00,04 ksoftirqd/0
  6 root     20  0      0  0    0 S  0,0  0,0  0:00,06 kworker/u256:0
  7 root     rt  0      0  0    0 S  0,0  0,0  0:00,02 migration/0
```

```
bigdata@localhost:~
Archivo Editar Ver Buscar Terminal Ayuda
top - 17:29:13 up 39 min,  3 users,  load average: 0,90, 0,33, 0,20
Tasks: 185 total,   1 running, 184 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0,2 us,  0,5 sy,  0,2 ni, 99,2 id,  0,0 wa,  0,0 hi,  0,0 si,  0,0 st
KiB Mem : 1867276 total, 150928 free, 1350108 used, 366240 buff/cache
KiB Swap: 2097148 total, 2097148 free,          0 used. 307180 avail Mem

PID USER    PR NI  VIRT  RES  SHR S %CPU %MEM TIME+ COMMAND
4157 bigdata  20  0 1969656 248484 50400 S  1,3 13,3  0:38,48 Web Conte...
4069 bigdata  20  0 2220848 234376 67064 S  1,0 12,6  0:27,39 firefox
1119 root     20  0 248308 39684 17512 S  0,3  2,1  0:15,57 Korg
4834 root     20  0 1784364 198460 19908 S  0,3 10,6  0:03,30 java
4974 root     20  0 1792780 175112 19960 S  0,3  9,4  0:03,60 java
  1 root     20  0 125304  3872 2416 S  0,0  0,2  0:00,91 systemd
  2 root     20  0      0  0    0 S  0,0  0,0  0:00,00 kthreadd
  3 root     20  0      0  0    0 S  0,0  0,0  0:00,04 ksoftirqd/0
```

Instalar Hadoop – Pseudo-Distributed

- Deberías poder acceder a la web del NameNode en <http://<ip>:50070>
- Importante: Si no accedes, es posible que el firewall esté activado
- Desactivar firewall: > service iptables stop

Welcome to CentOS | Instalar_Hadoop_singlen... | Namenode information

localhost:50070/dfshealth.html#tab-overview

67% | C | Buscar

Hadoop Overview Datanodes Datanode Volume Failures Snapshot Startup Progress Utilities

Browse the file system
Logs

Overview localhost:9000 (active)

Started: Thu Aug 05 17:28:42 +0200 2011

Version: 2.8.1, r20fe5304904fcf5a18053c389e43cd26f7a70fe

Compiled: Fri Jun 02 08:14:00 +0200 2017 by vinedv from branch-2.8.1.private

Cluster ID: CID-d94e0208-1e45-450f-b5fc-f08a8c496387

Block Pool ID: BP-176255935-127.0.0.1-1628176896475

Welcome to CentOS | Instalar_Hadoop_singlen... | Namenode information

localhost:50070/dfshealth.html#tab-overview

67% | C | Buscar

Namenode information - Mozilla Firefox

Welcome to CentOS | Instalar_Hadoop_singlen... | Namenode information

localhost:50070/dfshealth.html#tab-overview

67% | C | Buscar

Security is off.
Safemode is off.
1 files and directories, 0 blocks = 1 total filesystem object(s).
Heap Memory used 52.21 MB of 139.5 MB Heap Memory, Max Heap Memory is 889 MB.
Non Heap Memory used 31.42 MB of 46.44 MB Committed Non Heap Memory, Max Non Heap Memory is 214 MB.

Configured Capacity:	16.99 GB
DFS Used:	4 kB (0%)
Non DFS Used:	7.95 GB
DFS Remaining:	9.04 GB (53.21%)
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%
Live Nodes:	1 (Decommissioned: 0)
Dead Nodes:	0 (Decommissioned: 0)
Decommissioning Nodes:	0

Welcome to CentOS | Instalar_Hadoop_singlen... | Browsing HDFS

localhost:50070/explorer.html#

67% | C | Buscar

Browsing HDFS - Mozilla Firefox

Hadoop Overview Datanodes Datanode Volume Failures Snapshot Startup Progress Utilities

Browse Directory

Show 25 entries

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
No data available in table							

Showing 0 to 0 of 0 entries

Hadoop, 2017.

Instalar Hadoop – Pseudo-Distributed

- El servicio (demonio) de hdfs se empezará a ejecutar
 - top
 - ps aux | grep dfs
- free -m

```
top - 05:11:31 up 10:09, 3 users, load average: 0,29, 0,59, 0,47
Tasks: 180 total, 1 running, 179 sleeping, 0 stopped, 0 zombie
%Cpu(s): 2,0 us, 1,3 sy, 0,0 ni, 96,3 id, 0,0 wa, 0,0 hi, 0,3 si, 0,0 st
KiB Mem : 999920 total, 83988 free, 732924 used, 183008 buff/cache
KiB Swap: 2097148 total, 1680648 free, 416500 used. 59604 avail Mem

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
12124 bigdata 20 0 1521664 157980 5404 S 0,7 15,8 3:52.42 gnome-shell
31632 bigdata 20 0 2817228 125340 4888 S 0,7 12,5 0:09.10 java
31000 bigdata 20 0 2087230 114880 23240 S 0,7 11,3 0:12.37 firefox
1134 root 20 0 262776 45964 10988 S 0,3 4,6 1:39.24 Xorg
31786 bigdata 20 0 2801248 39700 4468 S 0,3 4,0 0:07.15 java
32061 root 20 0 0 0 0 S 0,3 0,0 0:00.28 kworker/0:2
1 root 20 0 193624 2132 560 S 0,0 0,2 0:10.41 systemd
2 root 20 0 0 0 0 S 0,0 0,0 0:01.79 kthreadd
3 root 20 0 0 0 0 S 0,0 0,0 0:02.10 ksoftirqd/0
7 root rt 0 0 0 0 S 0,0 0,0 0:00.00 migration/0
8 root 20 0 0 0 0 S 0,0 0,0 0:00.00 rcu_bh
9 root 20 0 0 0 0 S 0,0 0,0 0:04.39 rcu_sched
10 root rt 0 0 0 0 S 0,0 0,0 0:17.69 watchdog/0
12 root 20 0 0 0 0 S 0,0 0,0 0:00.00 kdevtmpfs
```

Instalar Hadoop – Pseudo-Distributed

Hadoop	Overview	Datanodes	Datanode Volume Failures	Snapshot	Startup Progress	Utilities
--------	----------	-----------	--------------------------	----------	------------------	-----------

Overview 'localhost:9000' (active)

Started:	Fri Jan 22 18:34:19 CET 2016
Version:	2.7.1, r15ecc87ccf4a0228f35af08fc56de536e6ce657a
Compiled:	2015-06-29T06:04Z by jenkins from (detached from 15ecc87)
Cluster ID:	CID-b6da5103-057a-41b4-9fd8-dadf83aabdd4
Block Pool ID:	BP-2111977582-172.16.150.129-1453483932990

Summary

Security is off.

Safemode is off.

1 files and directories, 0 blocks = 1 total filesystem object(s).

Heap Memory used 31.63 MB of 53.39 MB Heap Memory. Max Heap Memory is 966.69 MB.

Non Heap Memory used 30.54 MB of 31.94 MB Committed Non Heap Memory. Max Non Heap Memory is 214 MB.

Configured Capacity:

17.11 GB

Instalar Hadoop – Pseudo-Distributed

➤ Ahora creamos los directorios de usuarios en HDFS

- > bin/hdfs dfs -mkdir /user
- > bin/hdfs dfs -mkdir /user/root
- > bin/hdfs dfs -mkdir /user/bigdata

➤ Para probar

- > bin/hdfs dfs -put etc/hadoop/*.xml /user/bigdata
- ¿En qué directorio del HDFS se copian los ficheros?
- ¿Qué ocurre si no hubiéramos creado el directorio?

```
bigdata@localhost:/opt/hadoop
Archivo Editar Ver Buscar Terminal Ayuda
localhost: starting datanode, logging to /opt/hadoop-2.8.1/logs/hadoop-root-datanode-local
host.localdomain.out
Starting secondarynamenodes [0.0.0.0]
The authenticity of host '0.0.0.0 (0.0.0.0)' can't be established.
ECDSA key fingerprint is a0:d9:0f:06:4e:2b:0f:b7:31:44:c7:6d:0b:6c:53:1c.
Are you sure you want to continue connecting (yes/no)? yes
0.0.0.0: Warning: Permanently added '0.0.0.0' (ECDSA) to the list of known hosts.
0.0.0.0: starting secondarynamenode, logging to /opt/hadoop-2.8.1/logs/hadoop-root-secondarnamenode-localhost.localdomain.out
[root@localhost hadoop]#
[root@localhost hadoop]#
[root@localhost hadoop]#
[root@localhost hadoop]# bin/hdfs dfs -mkdir /user
[root@localhost hadoop]# bin/hdfs dfs -mkdir /user/root
[root@localhost hadoop]#
```

Browsing HDFS – Mozilla Firefox

Welcome to CentOS | Instalar_Hadoop_singlen... | Browsing HDFS

localhost:50070/explorer.html#/user 67% C Buscar

Hadoop Overview Datanodes Datanode Volume Failures Snapshot Startup Progress Utilities

Browse Directory

/user

Show 25 entries

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drw-r--r-	root	supergroup	0 B	Aug 05 17:41	0	0 B	bigdata
drw-r--r-	root	supergroup	0 B	Aug 05 17:40	0	0 B	root

Showing 1 to 2 of 2 entries

Browsing HDFS – Mozilla Firefox

Welcome to CentOS | Instalar_Hadoop_singlen... | Browsing HDFS

localhost:50070/explorer.html#/user/bigdata 67% C Buscar

Hadoop Overview Datanodes Datanode Volume Failures Snapshot Startup Progress Utilities

Browse Directory

/user/bigdata

Show 25 entries

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r-	root	supergroup	4.83 KB	Aug 05 20:04	1	128 MB	capacity-scheduler.xml
-rw-r--r-	root	supergroup	866 B	Aug 05 20:04	1	128 MB	core-site.xml
-rw-r--r-	root	supergroup	9.46 KB	Aug 05 20:04	1	128 MB	hadoop-policy.xml
-rw-r--r-	root	supergroup	849 B	Aug 05 20:04	1	128 MB	hdfs-site.xml
-rw-r--r-	root	supergroup	620 B	Aug 05 20:04	1	128 MB	https-site.xml

Instalar Hadoop – Pseudo-Distributed

➤ Para probar

- > bin/hdfs dfs -put etc/hadoop/*.xml /user/bigdata
- > bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.8.1.jar wordcount /user/root/prueba salidaHDFS1
- > bin/hdfs dfs -ls /user/root/salidaHDFS1
- > bin/hdfs dfs -cat /user/root/salidaHDFS1/part-r-00000

➤ Para parar todos los servicios

- # sbin/stop-dfs.sh

(No hacerlo de momento)

The screenshot shows the Hadoop Web UI interface. At the top, there's a navigation bar with links like Hadoop, Overview, Datanodes, Datanode Volume Failures, Snapshot, Startup Progress, and Utilities. Below the bar, the title is "Browse Directory" and the path is "/user/root/salidaHDFS1". A search bar and a "Go!" button are also present. The main content area is a table showing the directory contents:

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	root	supergroup	0 B	Aug 05 20:18	1	128 MB	_SUCCESS
-rw-r--r--	root	supergroup	10.16 KB	Aug 05 20:18	1	128 MB	part-r-00000

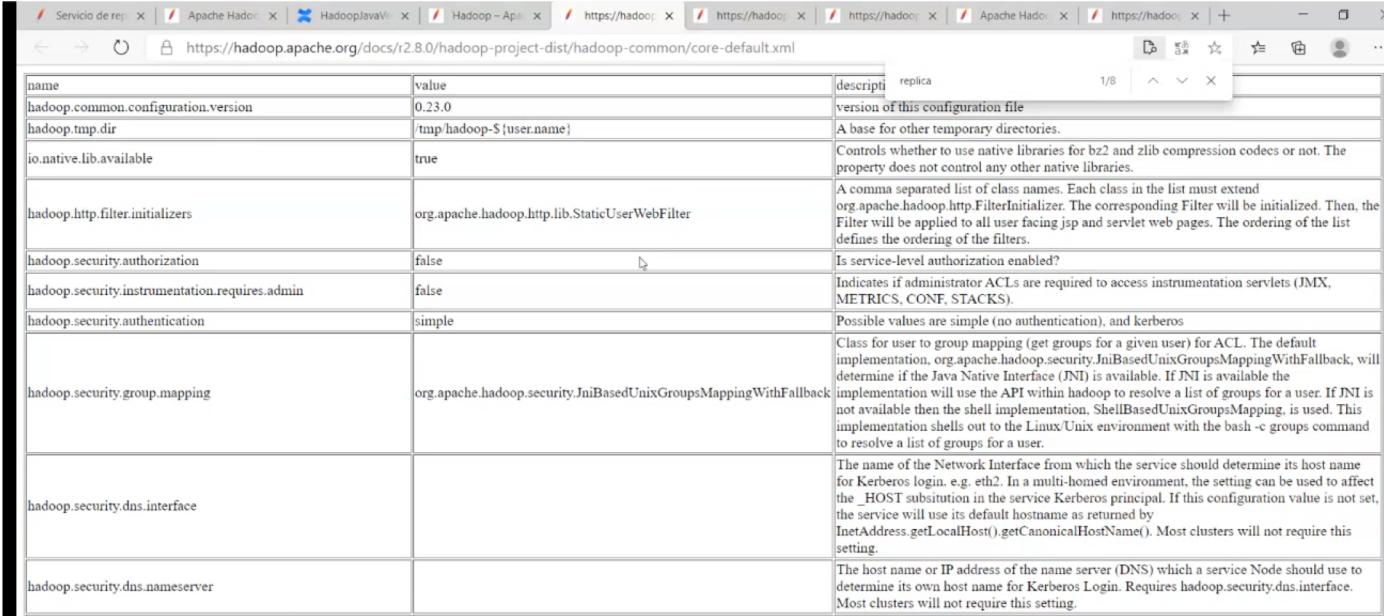
At the bottom of the table, it says "Showing 1 to 2 of 2 entries". There are "Previous" and "Next" buttons at the bottom right.

Instalar Hadoop – Pseudo-Distributed

- Para probar otro ejemplo y ver donde se crean los directorios
 - > bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.1.jar grep input output 'dfs[a-z.]+'
 - > bin/hdfs dfs -cat output/*

Instalar Hadoop – Pseudo-Distributed

- ¿Qué son todos estos ficheros de configuración que hemos editado? <https://hadoop.apache.org/docs/r2.8.0/>



The screenshot shows a web browser window displaying the configuration file `core-default.xml` from the Hadoop documentation at <https://hadoop.apache.org/docs/r2.8.0/>. The page lists various configuration properties with their values and descriptions.

name	value	description	replica
<code>hadoop.common.configuration.version</code>	0.23.0	version of this configuration file	1/8
<code>hadoop.tmp.dir</code>	<code>/tmp/hadoop-\$\{user.name\}</code>	A base for other temporary directories.	
<code>io.native.lib.available</code>	true	Controls whether to use native libraries for bz2 and zlib compression codecs or not. The property does not control any other native libraries.	
<code>hadoop.http.filter.initializers</code>	<code>org.apache.hadoop.http.lib.StaticUserWebFilter</code>	A comma separated list of class names. Each class in the list must extend <code>org.apache.hadoop.http.FilterInitializer</code> . The corresponding Filter will be initialized. Then, the Filter will be applied to all user facing jsp and servlet web pages. The ordering of the list defines the ordering of the filters.	
<code>hadoop.security.authorization</code>	false	Is service-level authorization enabled?	
<code>hadoop.security.instrumentation.requires.admin</code>	false	Indicates if administrator ACLs are required to access instrumentation servlets (JMX, METRICS, CONF, STACKS).	
<code>hadoop.security.authentication</code>	simple	Possible values are simple (no authentication), and kerberos	
<code>hadoop.security.group.mapping</code>	<code>org.apache.hadoop.security.JniBasedUnixGroupsMappingWithFallback</code>	Class for user to group mapping (get groups for a given user) for ACL. The default implementation, <code>org.apache.hadoop.security.JniBasedUnixGroupsMappingWithFallback</code> , will determine if the Java Native Interface (JNI) is available. If JNI is available the implementation will use the API within hadoop to resolve a list of groups for a user. If JNI is not available then the shell implementation, <code>ShellBasedUnixGroupsMapping</code> , is used. This implementation shells out to the Linux/Unix environment with the <code>bash -c groups</code> command to resolve a list of groups for a user.	
<code>hadoop.security.dns.interface</code>		The name of the Network Interface from which the service should determine its host name for Kerberos login, e.g. <code>eth2</code> . In a multi-homed environment, the setting can be used to affect the <code>_HOST</code> substitution in the service Kerberos principal. If this configuration value is not set, the service will use its default hostname as returned by <code>InetAddress.getLocalHost().getCanonicalHostName()</code> . Most clusters will not require this setting.	
<code>hadoop.security.dns.nameserver</code>		The host name or IP address of the name server (DNS) which a service Node should use to determine its own host name for Kerberos Login. Requires <code>hadoop.security.dns.interface</code> . Most clusters will not require this setting.	

Instalar Hadoop – Pseudo-Distributed

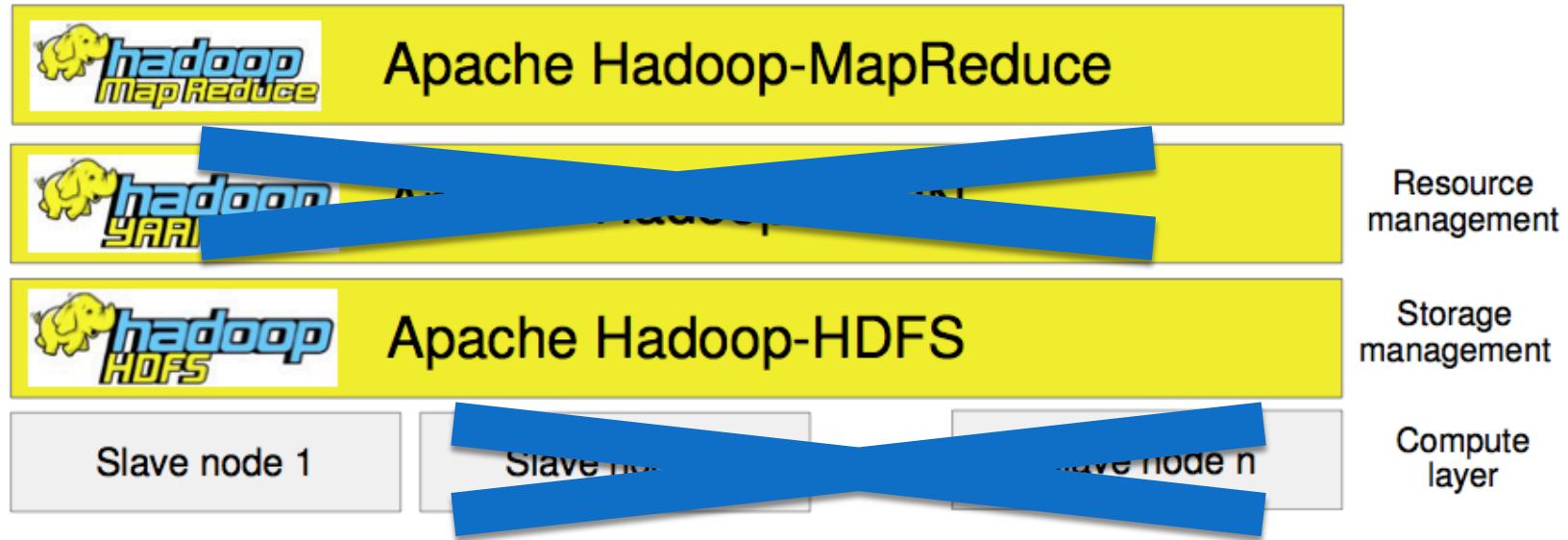
- Un pequeño resumen (<http://www.big-data.tips/hadoop-configuration>)
 - *The Hadoop configuration file **core-site.xml** contains pieces of information about the particular ‘Hadoop site’ itself. This includes the hostname and port number used for this particular Hadoop instance. Other optional information is the memory allocated for the file system. There can be also memory limits for storing data or more detailed configurations such as the size of read and write buffers.*
- Para investigar todos los parámetros disponibles:
 - <https://hadoop.apache.org/docs/r2.8.0/hadoop-project-dist/hadoop-common/core-default.xml>

Instalar Hadoop – Pseudo-Distributed

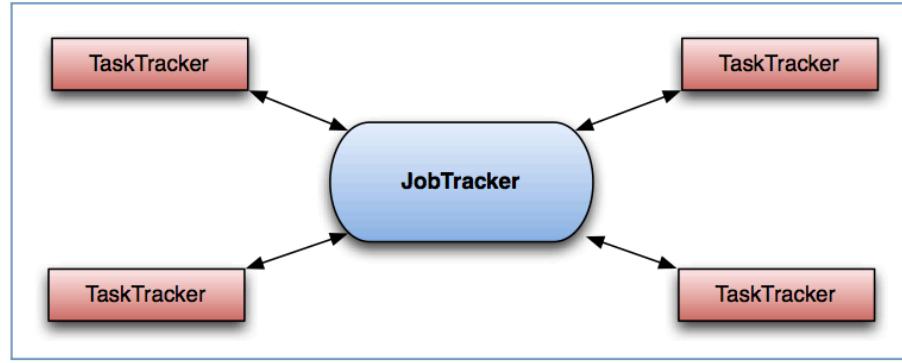
- Un pequeño resumen (<http://www.big-data.tips/hadoop-configuration>)
 - *The Hadoop configuration file **hdfs-site.xml** file contains information about the Hadoop Distributed File System (HDFS) that is part of the Hadoop distribution. It includes the value of ‘replication’ and the path to the ‘namenode’ as well as the paths to ‘datanodes’ based on the local file systems. This is needed in order to tell HDFS a concrete place where data in the Hadoop infrastructure is stored. Below is an example but needs to be configured according to your file system structure depending on the Hadoop infrastructure.*
- Para investigar todos los parámetros disponibles:
 - <https://hadoop.apache.org/docs/r2.8.0/hadoop-project-dist/hadoop-hdfs/hdfs-default.xml>

Instalación de Hadoop

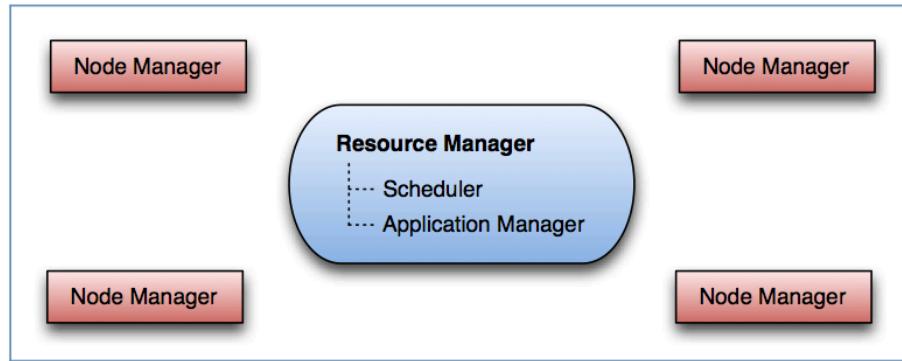
- Modo pseudo-distributed (hasta este punto)



MRv1 vs. MRv2 (Yarn)



- Un JobTracker (master) por cluster
- Cada esclavo ejecuta un TaskTracker



- Single Resource Manager por cluster
- Cada esclavo ejecuta un Node Manager

Imágenes obtenidas de Cloudera (<http://www.cloudera.com>)

Instalar Hadoop – Pseudo-Distributed con YARN

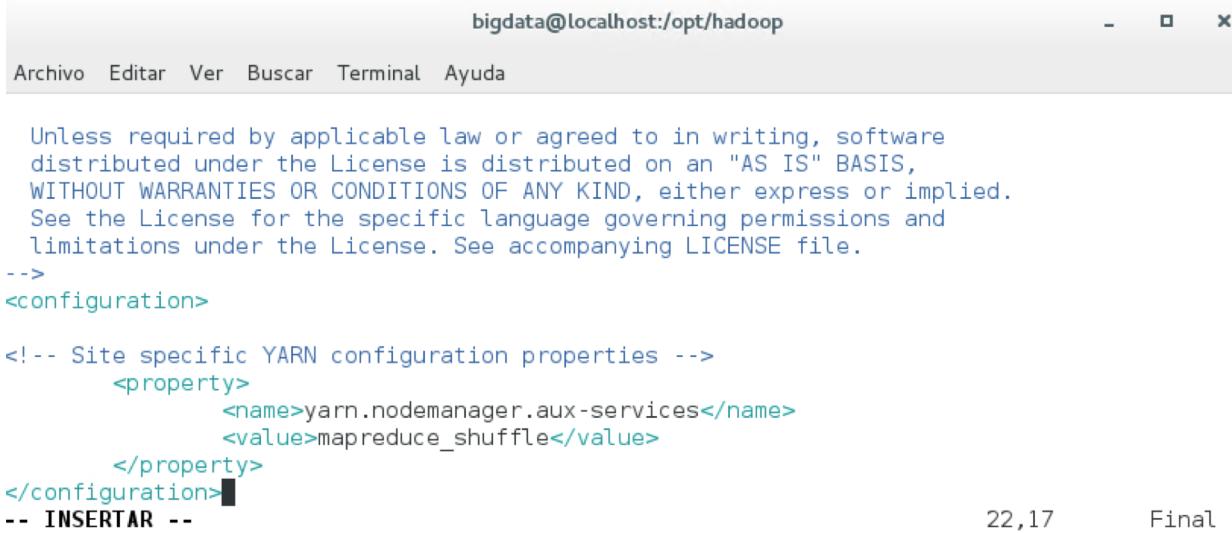
- La configuración anterior ejecuta tareas MapReduce usando MRv1. Sin embargo, es posible usar MRv2 (o YARN)
- Para ello es necesario ejecutar el servicio ResourceManager y NodeManager
- Partiendo de los pasos realizados antes... tras parar los servicios
- Añade la siguiente configuración a `etc/hadoop/mapred-site.xml`

```
<configuration>
    <property>
        <name>mapreduce.framework.name</name>
        <value>yarn</value>
    </property>
</configuration>
```

Instalar Hadoop – Pseudo-Distributed con YARN

- Añade la siguiente configuración a `etc/hadoop/yarn-site.xml`

```
<configuration>
    <property>
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
    </property>
</configuration>
```



A screenshot of a terminal window titled "bigdata@localhost:/opt/hadoop". The window contains the command-line history and the configuration code from the previous block.

```
bigdata@localhost:/opt/hadoop
Archivo Editar Ver Buscar Terminal Ayuda

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.
-->
<configuration>

<!-- Site specific YARN configuration properties -->
<property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
</property>
</configuration>
-- INSERTAR --
```

22,17

Final

Instalar Hadoop – Pseudo-Distributed con YARN

- Iniciar el ResourceManager y NodeManager
 - > sbin/start-yarn.sh (para finalizar sbin/stop-yarn.sh)
- Deberías poder acceder a la web del ResourceManager en <http://<ip>:8088>

```
[root@localhost hadoop]# vim etc/hadoop/yarn-site.xml
[root@localhost hadoop]# sbin/st
start-all.cmd      start-dfs.sh      stop-all.cmd      stop-dfs.sh
start-all.sh        start-secure-dns.sh stop-all.sh      stop-secure-dns.sh
start-balancer.sh   start-yarn.cmd   stop-balancer.sh stop-yarn.cmd
start-dfs.cmd       start-yarn.sh    stop-dfs.cmd     stop-yarn.sh
[root@localhost hadoop]# sbin/start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /opt/hadoop-2.8.1/logs/yarn-bigdata-resourcemanager-localhost.localdomain.out
localhost: starting nodemanager, logging to /opt/hadoop-2.8.1/logs/yarn-root-nodemanager-localhost.localdomain.out
[root@localhost hadoop]#
```

Se puede bloquear
si memoria > 2GB

The screenshot shows the Hadoop ResourceManager's web interface at localhost:8088/cluster. The title bar includes the Hadoop logo and the text "All Applications". The main content area displays cluster metrics, node metrics, and scheduler metrics. A sidebar on the left lists cluster nodes and applications, with a "Scheduler" section showing a table for capacity scheduling.

ID	User	Name	Application Type	Queue	Application Priority	StartTime	FinishTime	State	FinalStatus	Running Containers
Showing 0 to 0 of 0 entries										

Instalar Hadoop – Pseudo-Distributed con YARN

- Iniciar el ResourceManager y NodeManager
 - > sbin/start-yarn.sh (para finalizar sbin/stop-yarn.sh)
- Deberías poder acceder a la web del ResourceManager en <http://<ip>:8088>

The screenshot shows two windows. On the left is a terminal window titled 'root@localhost:/opt/hadoop' displaying log entries from a mapreduce job. On the right is a Mozilla Firefox browser window titled 'FINISHED Applications - Mozilla Firefox' showing the Hadoop ResourceManager's 'FINISHED Applications' page.

Terminal Output (root@localhost:/opt/hadoop):

```
Archivo Editar Ver Buscar Terminal Ayuda
21/08/10 20:59:35 INFO mapreduce.Job: map 63% reduce 0%
21/08/10 20:59:36 INFO mapreduce.Job: map 75% reduce 0%
21/08/10 20:59:43 INFO mapreduce.Job: map 88% reduce 0%
21/08/10 20:59:44 INFO mapreduce.Job: map 100% reduce 100%
21/08/10 20:59:46 INFO mapreduce.Job: Job job_1628621888157_0001
21/08/10 20:59:46 INFO mapreduce.Job: Counters: 49
  File System Counters
    FILE: Number of bytes read=20613
    FILE: Number of bytes written=1267850
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=27631
    HDFS: Number of bytes written=10406
    HDFS: Number of read operations=27
    HDFS: Number of large read operations=0
    HDFS: Number of write operations=2
  Job Counters
    Launched map tasks=8
    Launched reduce tasks=1
    Data-local map tasks=8
    Total time spent by all maps in occupied slots 0
    Total time spent by all reduces in occupied slots 0
    Total time spent by all map tasks (ms)=262047
    Total time spent by all reduce tasks (ms)=7673
    Total vcore-milliseconds taken by all map tasks 0
    Total vcore-milliseconds taken by all reduce tasks 0
```

Browser Window (FINISHED Applications - Mozilla Firefox):

The browser displays the 'FINISHED Applications' page. It shows cluster metrics (0 nodes, 1 application), active nodes (0), decommissioning nodes (0), and decommissioned nodes (0). The application table lists one completed job:

ID	User	Name	Application Type	Queue	Application Priority	StartTime	FinishTime	State
application_1628621888157_0001	root	word count	MAPREDUCE	default	0	Tue Aug 10 20:58:43 +0200 2021	Tue Aug 10 20:59:44 +0200 2021	FINISHED

Showing 1 to 1 of 1 entries

Se puede bloquear o ir lento si memoria <= 2GB

Instalar Hadoop – Pseudo-Distributed

- Un pequeño resumen (<http://www.big-data.tips/hadoop-configuration>)
 - *The Hadoop configuration file **mapred-site.xml** specifies which map-reduce framework is used that is in our example here YARN. Any Hadoop distribution contains a template of the ‘mapred-site.xml’ file named ‘mapred-site.xml.template’. We first copy this template file to the correct name and then add lines as shown below.*
- Para investigar todos los parámetros disponibles:
 - <https://hadoop.apache.org/docs/r2.8.0/hadoop-mapreduce-client/hadoop-mapreduce-client-core/mapred-default.xml>

Instalar Hadoop – Pseudo-Distributed

- Un pequeño resumen (<http://www.big-data.tips/hadoop-configuration>)
 - *The Hadoop configuration file **yarn-site.xml** is used for the Hadoop scheduling system ‘Yet Another Resource Negotiator (YARN)’. This component is also an integral part of Hadoop alongside HDFS.*
- Para investigar todos los parámetros disponibles:
 - <https://hadoop.apache.org/docs/r2.8.0/hadoop-yarn/hadoop-yarn-common/yarn-default.xml>

Instalar Hadoop – Pseudo-Distributed con YARN



Log

All Applications

Cluster	
About Nodes	
Node Labels	
Applications	
NEW	
NEW_SAVING	
SUBMITTED	
ACCEPTED	
RUNNING	
FINISHED	
FAILED	
KILLED	
Scheduler	
Tools	

Cluster Metrics															
Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	Vcores Used	Vcores Total	Vcores Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhea...	
0	0	0	0	0	0 B	8 GB	0 B	0	8	0	1	0	0	0	0
Scheduler Metrics															
Scheduler Type				Scheduling Resource Type				Minimum Allocation				Maximum Allocation			
Capacity Scheduler				[MEMORY]				<memory:1024, vCores:1>				<memory:8192, vCores:32>			
Show 20 entries															
ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress						
No data available in table															
Showing 0 to 0 of 0 entries															
First Previous															

Instalar Hadoop – Pseudo-Distributed

- Añadimos más servicios ejecutándose en segundo plano
 - top
 - ps aux | grep dfs
 - free -m

top - 05:11:31 up 10:09, 3 users, load average: 0,29, 0,59, 0,47										
Tasks: 180 total, 1 running, 179 sleeping, 0 stopped, 0 zombie										
%Cpu(s): 2,0 us, 1,3 sy, 0,0 ni, 96,3 id, 0,0 wa, 0,0 hi, 0,3 si, 0,0 st										
KiB Mem : 999920 total, 83988 free, 732924 used, 183008 buff/cache										
KiB Swap: 2097148 total, 1680648 free, 416500 used. 59604 avail Mem										
PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+ COMMAND
12124	bigdata	20	0	1521664	157980	5404	S	0,7	15,8	3:52.42 gnome-shell
31632	bigdata	20	0	2817228	125340	4888	S	0,7	12,5	0:09.10 java
31600	bigdata	20	0	2057250	114680	25240	S	0,7	11,3	0:12.97 firebox
1134	root	20	0	262776	45964	10988	S	0,3	4,6	1:39.24 Xorg
31786	bigdata	20	0	2801248	39700	4468	S	0,3	4,0	0:07.15 java
32061	root	20	0	6	6	0	S	0,3	0,0	0:06.20 kworker/0:2
1	root	20	0	193624	2132	560	S	0,0	0,2	0:10.41 systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:01.79 kthreadd
3	root	20	0	0	0	0	S	0,0	0,0	0:02.10 ksoftirqd/0
7	root	rt	0	0	0	0	S	0,0	0,0	0:00.00 migration/0
8	root	20	0	0	0	0	S	0,0	0,0	0:00.00 rcu_bh
9	root	20	0	0	0	0	S	0,0	0,0	0:04.39 rcu_sched
10	root	rt	0	0	0	0	S	0,0	0,0	0:17.69 watchdog/0
12	root	20	0	0	0	0	S	0,0	0,0	0:00.00 kdownmfs

Instalar Hadoop – Pseudo-Distributed con YARN

- Ahora podrías volver a lanzar la aplicación MapReduce de prueba de nuevo y ver su evolución en la web del Resource Manager
 - > bin/hdfs dfs -rm -r output
 - > bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.1.jar grep input output 'dfs[a-z.]+'
 - > bin/hdfs dfs -cat output/*
- O si echas de menos el ‘wordcount’...
 - > bin/hdfs dfs -rm -r output
 - > bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.1.jar wordcount input output
 - > bin/hdfs dfs -cat output/*

Instalar Hadoop – Pseudo-Distributed con YARN

- La aplicación debe aparecer en la web...



Log

All Applications

Cluster	
About	
Nodes	
Node Labels	
Applications	
NEW	
NEW_SAVING	
SUBMITTED	
ACCEPTED	
RUNNING	
FINISHED	
FAILED	
KILLED	
Scheduler	
Tools	

Cluster Metrics																		
Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	Vcores Used	Vcores Total	Vcores Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhea...				
1	0	1	0	7	8 GB	8 GB	0 B	7	8	0	1	0	0	0	0	0		
Scheduler Metrics																		
Scheduler Type				Scheduling Resource Type				Minimum Allocation				Maximum Allocation						
Capacity Scheduler				[MEMORY]				<memory:1024, vCores:1>				<memory:8192, vCores:8>						
Show 20 entries																		
ID				User	Name	Application Type		Queue	StartTime	FinishTime	State	FinalStatus	Progress	Ap...				
application_1453722155914_0001				root	grep-search	MAPREDUCE		default	Mon Jan 25 12:45:48 +0100 2016	N/A	RUNNING	UNDEFINED		Ap...				
Showing 1 to 1 of 1 entries																		
First Previous																		

Instalar Hadoop – Pseudo-Distributed con YARN



Logged in as: dr.who

Application application_1453722155914_0001

Kill Application

User: root
Name: grep-search
Application Type: MAPREDUCE
Application Tags:
YarnApplicationState: RUNNING: AM has registered with RM and started running.
FinalStatus Reported by AM: Application has not completed yet.
Started: lun ene 25 12:45:48 +0100 2016
Elapsed: 47sec
Tracking URL: ApplicationMaster
Diagnostics:

Application Metrics

Total Resource Preempted: <memory:0, vCores:0>
Total Number of Non-AM Containers Preempted: 0
Total Number of AM Containers Preempted: 0
Resource Preempted from Current Attempt: <memory:0, vCores:0>
Number of Non-AM Containers Preempted from Current Attempt: 0
Aggregate Resource Allocation: 280848 MB-seconds, 226 vcore-seconds

Show 20 entries

Attempt ID	Started	Node	Logs
appattempt_1453722155914_0001_000001	Mon Jan 25 12:45:48 +0100 2016	http://hadoop-master:8042	Logs

Showing 1 to 1 of 1 entries

Search:

First Previous 1 Next Last

Instalar Hadoop – Pseudo-Distributed con YARN

Cluster

- About
- Nodes
- Node Labels
- Applications
 - NEW
 - NEW_SAVING
 - SUBMITTED
 - ACCEPTED
 - RUNNING
 - FINISHED
 - FAILED
 - KILLED
- Scheduler

Tools

Application Attempt State: RUNNING

AM Container: container_1453722155914_0001_01_000001

Node: hadoop-master:51320

Tracking URL: ApplicationMaster

Diagnostics Info:

Application Attempt Metrics

Application Attempt Headroom : <memory:0, vCores:0>

Total Allocated Containers: 7

Each table cell represents the number of NodeLocal/RackLocal/OffSwitch containers satisfied by NodeLocal/RackLocal/OffSwitch resource requests.

	Node Local Request	Rack Local Request	Off Switch Request
Num Node Local Containers (satisfied by)	6		
Num Rack Local Containers (satisfied by)	0	0	
Num Off Switch Containers (satisfied by)	0	0	1

Total Outstanding Resource Requests: <memory:23552, vCores:23>

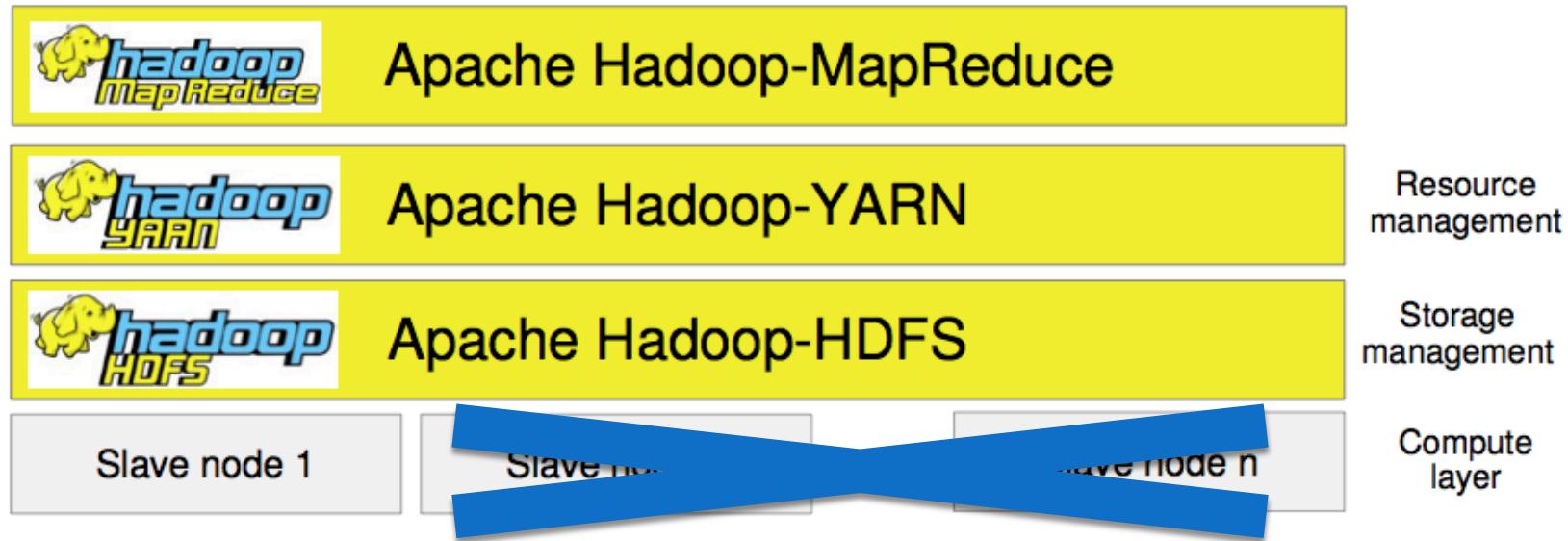
Priority	ResourceName	Capability	NumContainers	RelaxLocality	NodeLabelExpression
20	hadoop-master	<memory:1024, vCores:1>	23	true	
20	*	<memory:1024, vCores:1>	23	true	
20	/default-rack	<memory:1024, vCores:1>	23	true	

Show 20 entries Search:

Container ID	Node	Container Exit Status	Logs
container_1453722155914_0001_01_000007	http://hadoop-master:8042	0	Logs
container_1453722155914_0001_01_000006	http://hadoop-master:8042	0	Logs
container_1453722155914_0001_01_000005	http://hadoop-master:8042	0	Logs
container_1453722155914_0001_01_000004	http://hadoop-master:8042	0	Logs
container_1453722155914_0001_01_000003	http://hadoop-	0	Logs

Instalación de Hadoop

- Modo pseudo-distributed (ahora)



Instalación de Hadoop

➤ Instalación *Fully Distributed*

- Los demonios de Hadoop se ejecutan en un cluster de máquinas
- HDFS se emplea para distribuir datos entre todos los nodos
- A menos que se emplee un cluster pequeño (menos de 10 o 20 nodos), el NameNode y JobTracker deben ejecutarse en nodos dedicados
 - Para pequeños clusters pueden ejecutarse en el mismo nodo