

Hadoop

Luciano Olmedo Gerardo Lareu
Leandro Pablo Remorini

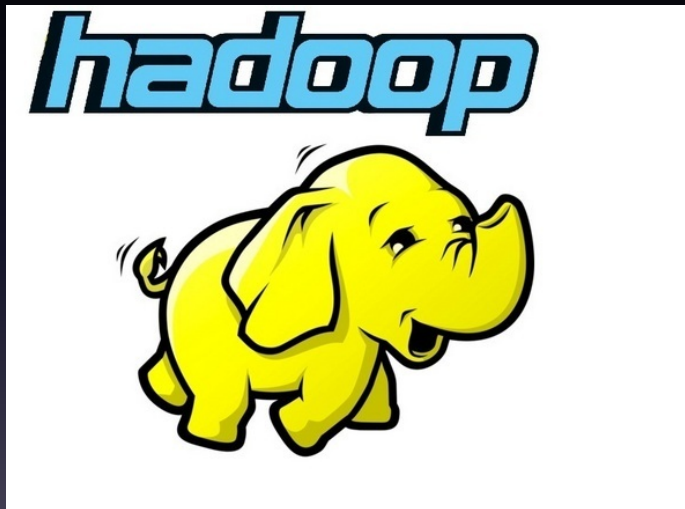
-

Prof : Jose Luis Di Biase

UNQ

Junio 25, 2015

Hadoop



(a)

Hadoop

Temario

Hadoop

Temario

- Que es Hadoop

Hadoop

Temario

- Que es Hadoop
- Hadoop en detalle

Hadoop

Temario

- Que es Hadoop
- Hadoop en detalle
- Que vamos a hacer con Hadoop

Que es Hadoop?

(Apache) Hadoop es un framework de software libre con una licencia Apache Commons que consiste basicamente de:

Que es Hadoop?

(Apache) Hadoop es un framework de software libre con una licencia Apache Commons que consiste basicamente de:

- Un sistema de archivos distribuido, escalable (linealmente) escrito en Java (HDFS).

Que es Hadoop?

(Apache) Hadoop es un framework de software libre con una licencia Apache Commons que consiste basicamente de:

- Un sistema de archivos distribuido, escalable (linealmente) escrito en Java (HDFS).
- Algo que me permita trabajar de forma distribuida con muchos datos en clusters de computadoras. (MapReduce)

Que es Hadoop?

(Apache) Hadoop es un framework de software libre con una licencia Apache Commons que consiste basicamente de:

- Un sistema de archivos distribuido, escalable (linealmente) escrito en Java (HDFS).
- Algo que me permita trabajar de forma distribuida con muchos datos en clusters de computadoras. (MapReduce)
- YARN: Un framework para gestionar los recursos del cluester y los job scheduling.

Que es Hadoop?

(Apache) Hadoop es un framework de software libre con una licencia Apache Commons que consiste basicamente de:

- Un sistema de archivos distribuido, escalable (linealmente) escrito en Java (HDFS).
- Algo que me permita trabajar de forma distribuida con muchos datos en clusters de computadoras. (MapReduce)
- YARN: Un framework para gestionar los recursos del cluester y los job scheduling.
- Hadoop Common: Utilidades que soportan a los otros modulos de Hadoop.

Un poco (muy poco) de historia...

La idea detras de Hadoop surgio en los documentos Google para MapReduce y Google File System (GFS).

Un poco (muy poco) de historia...

La idea detras de Hadoop surgio en los documentos Google para MapReduce y Google File System (GFS).

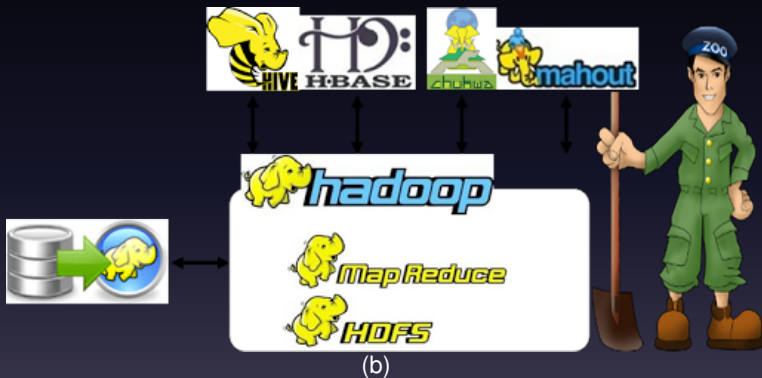
- Google puso los papers, mucha gente puso el codigo, Yahoo (entre otros) financiamiento.

Un poco (muy poco) de historia...

La idea detras de Hadoop surgio en los documentos Google para MapReduce y Google File System (GFS).

- Google puso los papers, mucha gente puso el codigo, Yahoo (entre otros) financiamiento.
- Trata de un sistema base con agregados

Un poco de Arquitectura



Un poco de Arquitectura

Hadoop posee varias formas de ejecución:

Un poco de Arquitectura

Hadoop posee varias formas de ejecución:

- Modo Standalone: No se necesita configurar casi nada.

Un poco de Arquitectura

Hadoop posee varias formas de ejecución:

- Modo Standalone: No se necesita configurar casi nada.
- Modo PseudoDistribuido: Un sistema basado en cliente servidor, pero que se ejecuta en modo local todo. O sea sirve para simular un cluster de varios nodos pero corriendo en una misma máquina es decir en la misma Java VM.

Un poco de Arquitectura

Hadoop posee varias formas de ejecución:

- Modo Standalone: No se necesita configurar casi nada.
- Modo PseudoDistribuido: Un sistema basado en cliente servidor, pero que se ejecuta en modo local todo. O sea sirve para simular un cluster de varios nodos pero corriendo en una misma máquina es decir en la misma Java VM.
- Modo distribuido: Infraestructura completa con varios nodos de almacenamiento, ejecución.No vamos a focalizar en esto

HDFS

HDFS es una implementación GFS. Se trata de un sistema de archivos distribuido en cada nodo del cluster. Utiliza un tamaño de bloque variable (64Mb, 128Mb, 256, ...) y está pensado para trabajar con grandes ficheros de datos. Está diseñado para ser un sistema de ficheros distribuido, escalable, tolerante a fallos, con un intensivo acceso a datos y alta concurrencia.

HDFS

HDFS es una implementación GFS. Se trata de un sistema de archivos distribuido en cada nodo del cluster. Utiliza un tamaño de bloque variable (64Mb, 128Mb, 256, ...) y está pensado para trabajar con grandes ficheros de datos. Está diseñado para ser un sistema de ficheros distribuido, escalable, tolerante a fallos, con un intensivo acceso a datos y alta concurrencia.

- Sigue una filosofía "Write once, read more" en concreto para albergar grandes ficheros y principalmente pensado para programas batch donde los datos no son en tiempo real.

HDFS

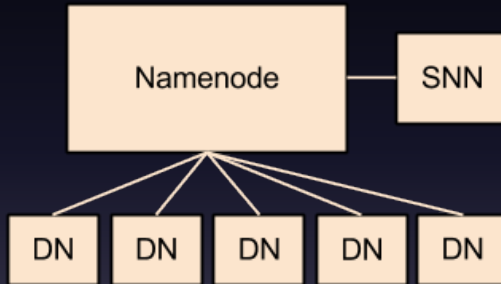
HDFS es una implementación GFS. Se trata de un sistema de archivos distribuido en cada nodo del cluster. Utiliza un tamaño de bloque variable (64Mb, 128Mb, 256, ...) y está pensado para trabajar con grandes ficheros de datos. Está diseñado para ser un sistema de ficheros distribuido, escalable, tolerante a fallos, con un intensivo acceso a datos y alta concurrencia.

- Sigue una filosofía "Write once, read more" en concreto para albergar grandes ficheros y principalmente pensado para programas batch donde los datos no son en tiempo real.
- Divide archivos en bloques de tamaño fijo y los distribuye en los distintos nodos del cluster. Debido a que se produce mucho tráfico por la red para distribuir la información se opta por un tamaño de bloque alto.

HDFS

La gestión del cluster la realiza el NameNode. Es el nodo maestro encargado de gestionar los metadatos de los ficheros y los directorios, coordina los bloques que se envían a cada datanode (quienes hacen el trabajo y donde se guarda la información) monitorizando su estado para asegurar que todos los trabajos se completan correctamente, redirigiendo bloques a otros datanodes si alguno estuviera caído. La información de los metadatos se almacena en memoria RAM para que el acceso sea más rápido.

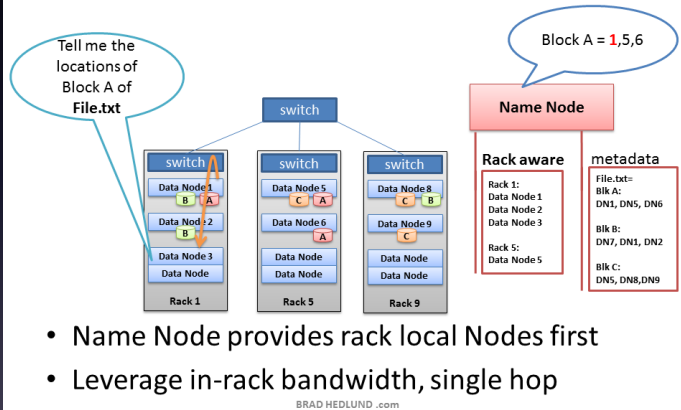
HDFS



(c)

Leer datos de HDFS

Data Node reading files from HDFS

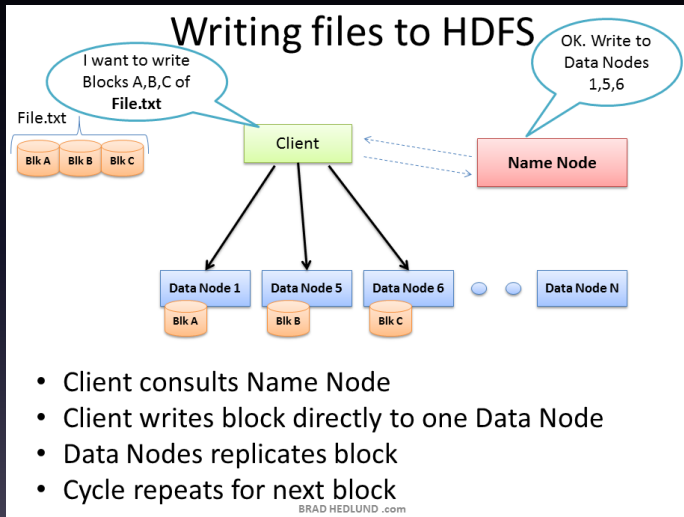


- Name Node provides rack local Nodes first
- Leverage in-rack bandwidth, single hop

BRAD HEDLUND .com

(d)

Escribir datos de HDFS



(e)

Replicación

La alta disponibilidad de los datos en Hadoop es posible debido a que lleva implícita la replicación de datos en un clúster Hadoop:

Replicación

La alta disponibilidad de los datos en Hadoop es posible debido a que lleva implícita la replicación de datos en un clúster Hadoop:

- Un bloque de archivo se replica en varios "nodos de datos" en función del "factor de replicación" del clúster Hadoop, que podría ser 1, 2, 3...

Replicación

La alta disponibilidad de los datos en Hadoop es posible debido a que lleva implícita la replicación de datos en un clúster Hadoop:

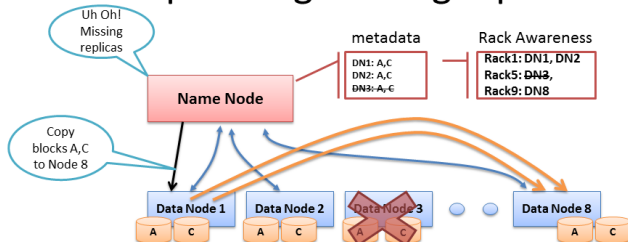
- Un bloque de archivo se replica en varios "nodos de datos" en función del "factor de replicación" del clúster Hadoop, que podría ser 1, 2, 3...
- Un factor de replicación de 1 indica que un bloque de archivo residirá en un sólo "nodo de datos".

Replicación

La alta disponibilidad de los datos en Hadoop es posible debido a que lleva implícita la replicación de datos en un clúster Hadoop:

- Un bloque de archivo se replica en varios "nodos de datos" en función del "factor de replicación" del clúster Hadoop, que podría ser 1, 2, 3...
- Un factor de replicación de 1 indica que un bloque de archivo residirá en un sólo "nodo de datos".
- Un factor de replicación de 2 indica que un bloque de archivo residirá en dos "nodos de datos", ya sea dentro del mismo rack o en uno que esté físicamente a miles de kilómetros de distancia; etc, etc.

Re-replicating missing replicas



- Missing Heartbeats signify lost Nodes
- Name Node consults metadata, finds affected data
- Name Node consults Rack Awareness script
- Name Node tells a Data Node to re-replicate

BRAD HEDLUND .com

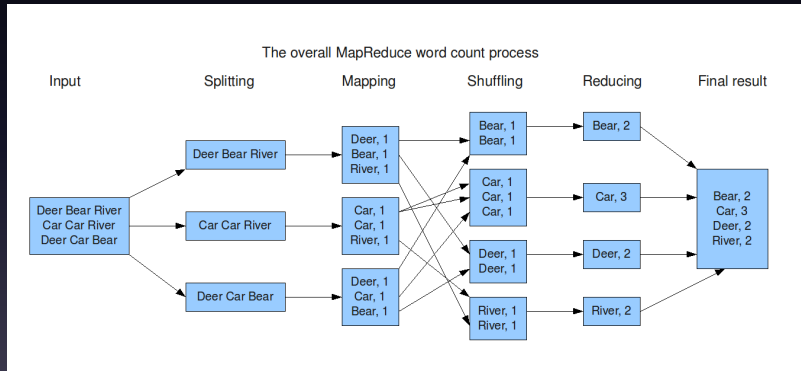
(f)

MapReduce : Map y Fold en forma de fichas



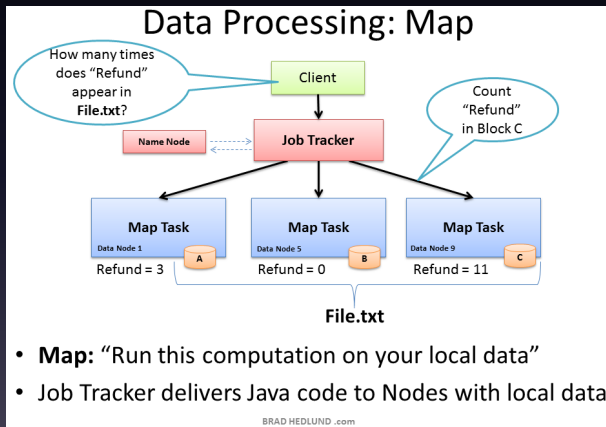
(g) Alf

MapReduce : Map y Fold en forma de fichas



(h) MapReduce

MapReduce : Map y Fold en forma de fichas



BRAD HEDLUND .com

(i) MapReduce

Las otras herramientas: PIG, HIVE ,

...



(j)

Apache Pig, inicialmente desarrollado por Yahoo, permite a los usuarios de Hadoop centrarse más en el análisis de los datos y menos en la creación de programas MapReduce. Pig consta de dos componentes:

- El lenguaje en si, llamado PigLatin.

Apache Pig, inicialmente desarrollado por Yahoo, permite a los usuarios de Hadoop centrarse más en el análisis de los datos y menos en la creación de programas MapReduce. Pig consta de dos componentes:

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

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

HBASE

HBase, se trata de la base de datos de Hadoop. HBase es el componente de Hadoop a usar, cuando se requiere escrituras/lecturas en tiempo real y acceso aleatorio para grandes conjuntos de datos. Es una base de datos NOSQL del tipo clave-valor.

Que vamos a hacer con Hadoop

Lo que hicimos:

Que vamos a hacer con Hadoop

Lo que hicimos:

- Instalación y configuración.

Que vamos a hacer con Hadoop

Lo que hicimos:

- Instalación y configuración.
- Correr el sistema de archivo

Implementación de HADOOP

Especificaciones del sistema de prueba

Implementación de HADOOP

Especificaciones del sistema de prueba

- Ubuntu 14.04

Implementación de HADOOP

Especificaciones del sistema de prueba

- Ubuntu 14.04
- Interl dualcore 4400

Implementación de HADOOP

Especificaciones del sistema de prueba

- Ubuntu 14.04
- Interl dualcore 4400
-

Implementación de HADOOP

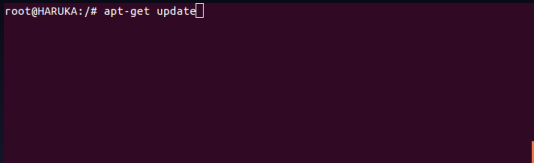
Requerimientos previos

Implementación de HADOOP

Requerimientos previos

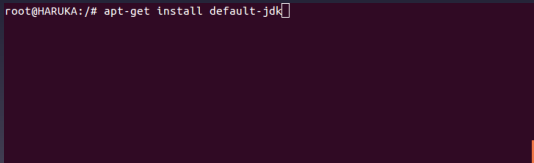
- Instalación de java

```
root@HARUKA:/# apt-get update
```



(m)

```
root@HARUKA:/# apt-get install default-jdk
```



(n)

Implementacion de HADOOP

Hadoop tambien requiere la intalacion de ssh y rsync

Implementación de HADOOP

Hadoop también requiere la instalación de ssh y rsync

- Instalación de ssh

```
root@HARUKA:/# apt-get install ssh
```

(q)

- Instalación de rsync

```
root@HARUKA:/# apt-get install rsync
```

(r)

Implementacion de HADOOP

Ya tendriamos preparado el sistema para instalar Hadoop

Implementacion de HADOOP

Ya tendríamos preparado el sistema para instalar Hadoop

- Descargamos la version 2.7.0 de Hadoop de la pagina <http://hadoop.apache.org> y la descoprimimos. Nos tiene que quedar una carpeta /hadoop con la que vamos a trabajar en adelante

hadoop.apache.org/releases.html#06+July%2C+2015%3A+Release+2.7.1+%28stable%29+available

hadoop

Apache Hadoop Releases

Download

Hadoop is released as source code tarballs with corresponding binary tarballs for convenience. The downloads are distributed via mirror sites and or SHA-256.

Version	Release Date	Tarball	Signature	GPG
2.7.1	06 July, 2015	source	signature	53F301C457AD8FD...
		binary	signature	991DC34E_A42A80B2...
2.6.0	18 Nov, 2014	source	signature	51A6520B_962D7D31...
		binary	signature	7A7E24E7_F569AFC2...
2.5.2	19 Nov, 2014	source	signature	139EF872_0F5637E...
		binary	signature	08DB4850_A38220B...

To verify Hadoop releases using GPG:

(t)

Implementación de HADOOP

Configurando Hadoop

Implementacion de HADOOP

Configurando Hadoop

- En primer lugar tenemos que obtener la ruta de la instalacion de java. Esto lo logramos con el siguiente comando

```
root@HARUKA:/# update-alternatives --config java
Sólo hay una alternativa en el grupo de enlaces java (provee /usr/bin/java): /usr/lib/jvm/java-7-openjdk-1386/jre/bin/java
Nada que configurar.
root@HARUKA:/#
```

(v)

En este caso la ruta seria /usr/lib/jvm/java-7-openjdk-i386

Implementación de HADOOP

Configurando Hadoop

Implementacion de HADOOP

Configurando Hadoop

- Ahora debemos editar el archivo `hadoop/etc/hadoop-env.sh`. Deberia quedar asi.

```
# The only required environment variable is JAVA_HOME. All others are
# 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/java-7-openjdk-1386

# 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 of
# data transfer protocol using non-privileged ports.
```

(X)

Implementacion de HADOOP

Configurando Hadoop

Implementación de HADOOP

Configurando Hadoop

- Hadoop utiliza ssh para conectarse a los nodos. Ssh normalmente requiere de un usuario y password para ingresar. Esto es necesario eliminarlo para acceder directamente. Esto lo logramos con el siguiente comando.

```
root@HARUKA:/# ssh-keygen -t rsa -P ''
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
/root/.ssh/id_rsa already exists.
Overwrite (y/n)? y
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:
02:9b:00:f5:99:f0:e8:d6:53:cf:76:f0:f8:a5:65:9b root@HARUKA
The key's randomart image is:
+--[ RSA 2048 ]-----+
|..o
| . = o
|  o * . .
| . o = o +
|  o = . S o +
| . . o o = o
|          o E
+-----+
root@HARUKA:/# cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
root@HARUKA:/#
```

(Z)

Implementacion de HADOOP

Configurando Hadoop

Implementacion de HADOOP

Configurando Hadoop

- Probamos Hadoop con el siguiente comando. Se deberia ver algo asi.

```
root@HARUKA:/home/whish/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
  credential         interact with credential providers
  Hadoop jar and the required libraries
  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@HARUKA:/home/whish/hadoop#
```

()

Preguntas?