



# Introducción al uso del clúster de Big Data de la UO.

**Autor: Robert Amaurys Reina Valladares.**

Dirección de Informatización, Universidad de Oriente (UO).  
Departamento de Informática, Facultad de Ingeniería en Telecomunicaciones, Informática y Biomédica, UO.  
Mar. 2020 / Santiago de Cuba.

# INTRODUCCIÓN

## Elevado incremento de los datos almacenados electrónicamente:

- En 2013 el universo digital se estimó en unos 4.4 zettabyte y para 2020 se espera la cifra de 44 zettabyte.
  - 1 zettabyte = 1 billón de terabyte.
- Facebook hospeda 240 billones de fotos (2014), crece a razón de 7 petabytes por mes.
- Fuente: White, Tom. “Hadoop: The Definitive Guide”. 4ta Edición. Abril, 2015.

# INTRODUCCIÓN

## Almacenamiento y análisis de los datos (1):

- La velocidad de transferencia de un HDD de 1TB (norma) está alrededor de los 100 MB/s, su lectura total demora unas 2 horas.
  - La escritura puede ser un poco más lenta.
- Una manera obvia de reducir el tiempo es leer de múltiples HDD al mismo tiempo.
  - Con 100 HDD en paralelo la lectura anterior puede hacerse en menos de 2 minutos.

# INTRODUCCIÓN

## **Almacenamiento y análisis de los datos (2):**

- El análisis paralelo de los datos necesita soportar tolerancia al fallo del hardware.
  - Una manera común de solución la pérdida de datos es la replicación. Redundancia de datos.
- El otro reto es la necesidad de combinar los datos en tiempo de análisis.
  - La lectura de un disco puede necesitar ser combinada con la de cualquiera de los restantes 99 discos – siguiendo el ejemplo anterior.

## Definición:

- El término Big Data hace referencia al almacenamiento de grandes volúmenes de datos de una gran variedad (estructurados y no estructurados), para su posterior procesamiento y análisis a gran velocidad con el fin de obtener información de valor intrínseco.
- Big data es una colección de datos grande, complejos, muy difícil de procesar a través de herramientas de gestión y procesamiento de datos tradicionales.

## Casos de uso:

- Análisis de comportamiento.
- Segmentación de clientes.
- Soporte predictivo.
- Análisis de mercado y optimización de precios.
- Predicción de amenazas.
- Detección de fraudes.
- ...

## Tratamiento computacional:

- Todos los modelos de programación diseñados para el análisis de Big Data se basan necesariamente en técnicas de paralelismo masivo.
- MapReduce es el modelo más utilizado.
  - Desarrollado por Google para procesar y generar grandes volúmenes de datos de forma paralela.
- MapReduce ha sido ampliamente adoptado gracias a su implementación de software libre Apache Hadoop.

# HADOOP

## Hadoop (1):

- Proyecto de software libre, gestionado inicialmente por Yahoo! y actualmente por Apache Software Foundation.
- Implementa el modelo de programación MapReduce basándose en la investigación que publicó Google sobre MapReduce y Google File System (GFS) en 2004.
  - Escrito en Java.
- Permite procesar grandes volúmenes de datos de forma distribuida, fiable y escalable.
  - Maneja miles de nodos y Petabytes de datos.

# HADOOP

## Hadoop (2):

- Mundialmente utilizado por innumerables empresas para ejecutar cálculos sobre enormes cantidades de datos de forma distribuida a través de miles de nodos.
  - A9.com, eBay, Facebook, Fox Interactive Media, IBM, ImageShack, Last.fm, LinkedIn, Twitter, MercadoLibre, The New York Times, entre otros.

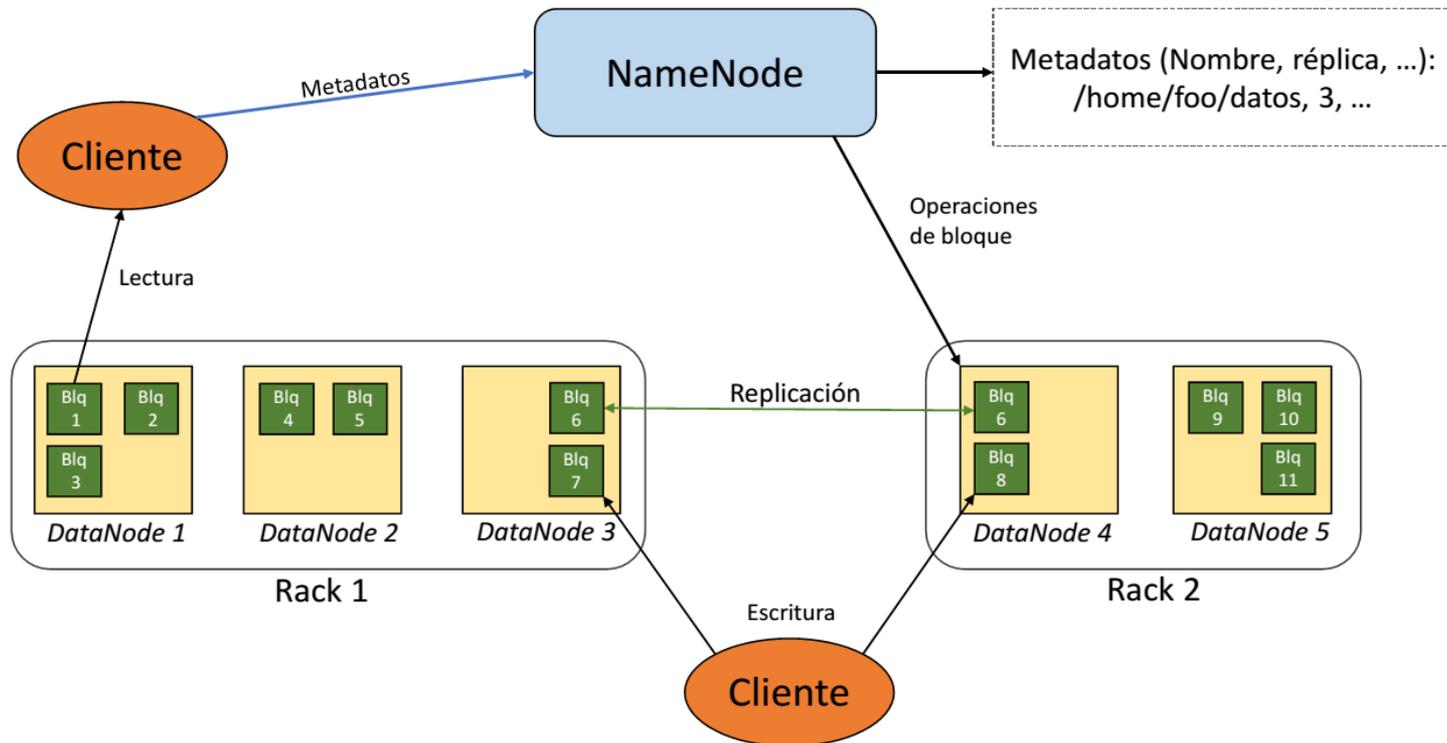
# HADOOP

## Módulos:

- Hadoop Common.
  - Utilidades más comunes para los otros módulos.
- Hadoop Distributed File System (HDFS):
  - Sistema distribuido de ficheros que proporciona acceso de alto rendimiento a los datos de aplicación.
- Hadoop YARN.
  - Framework de planificación de tareas y gestión de recursos de un clúster.
- Hadoop MapReduce.
  - Implementación del modelo MapReduce basado en YARN.

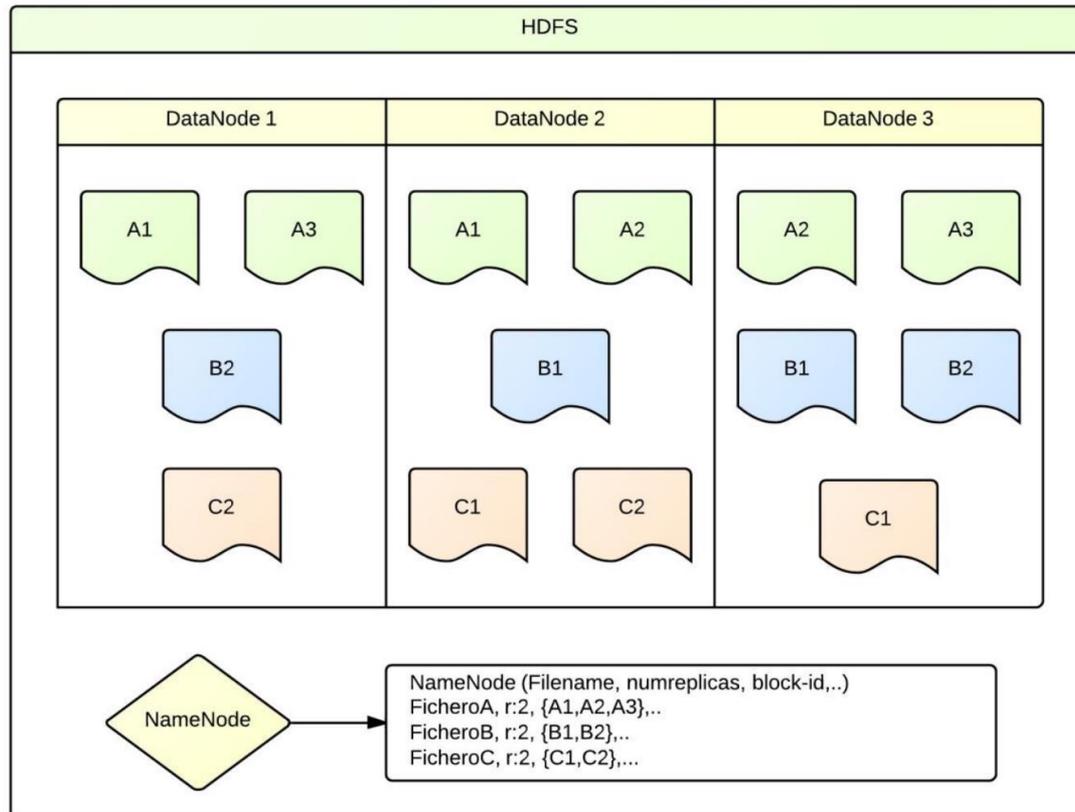
# HADOOP - HDFS

## Arquitectura de HDFS:



# HADOOP - HDFS

## Replicación y división de ficheros:



## **YARN (Yet Another Resource Negotiator):**

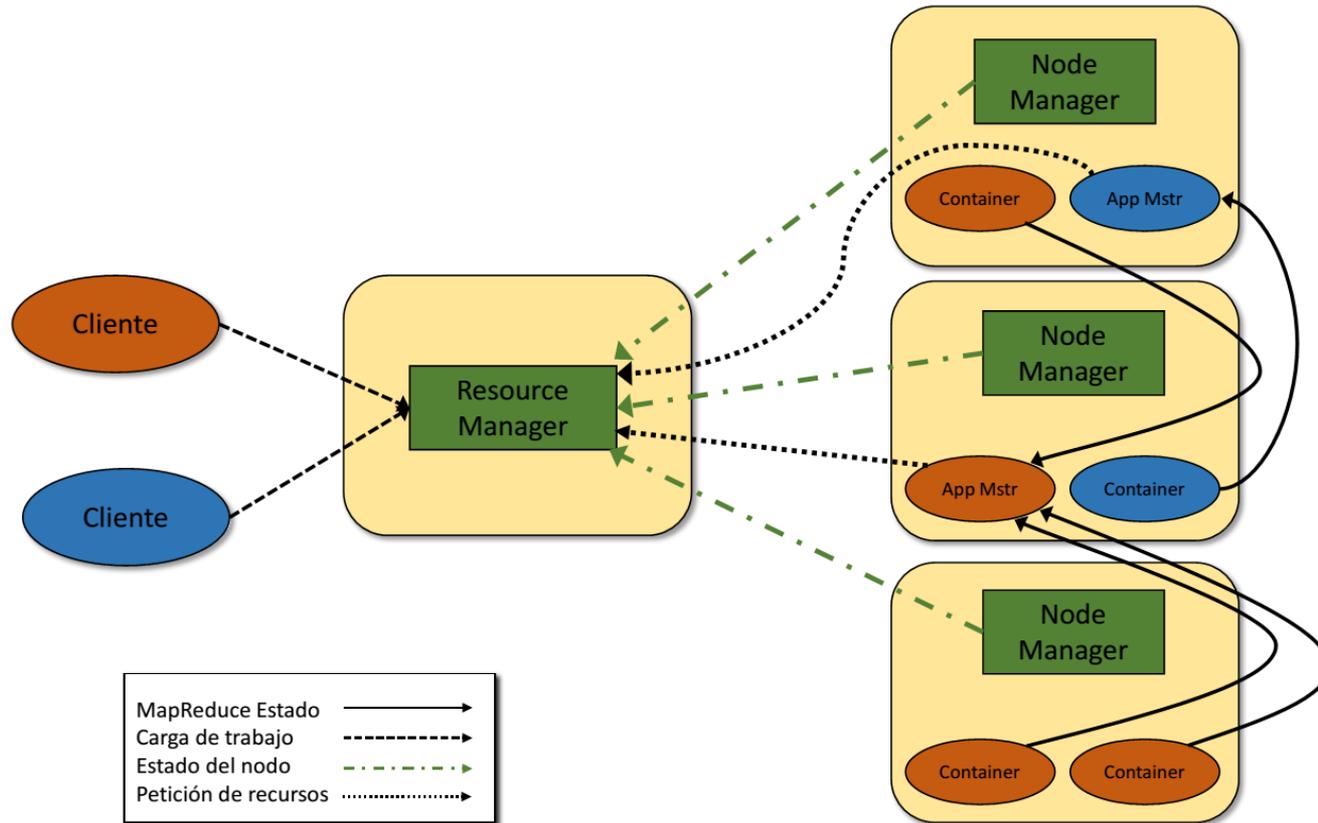
- Es esencialmente un sistema para administrar aplicaciones distribuidas.
- El ResourceManager es el componente central que arbitra todos los recursos disponibles del clúster.
- Un NodeManager por nodo que recibe y toma las decisiones recibidas desde el ResourceManager,
  - Es el responsable de gestionar los recursos disponibles del nodo donde se ejecuta.

## **YARN (Yet Another Resource Negotiator):**

- ApplicationMaster para cada una de la aplicaciones que ejecuta.
  - Responsable de negociar los recursos que necesita su aplicación y monitorizar su estado.

# HADOOP - YARN

## Arquitectura y flujo de trabajo de YARN:



# CLÚSTER BIGDATA DE LA UO

## Particulares:

- Parte de la red académica de supercómputo de Cuba.
  - Formada por 4 centros (UCLV, UO, UCI, ETI-BioCubaFarma).
  - Apoyada por proyectos internacionales (VLIR).
  - Basada en tecnologías libres (homogéneas).



- Software base - Hortonworks Data Platform 3.1.
  - 100% free and Open Source.
  - Apache Licensed software.
  - Basada en Apache Hadoop.



# RECURSOS DEL CLÚESTER

## Hardware:

- Nodos de acceso (ssh - puerto 22)

Nombre de dominio	Cores	RAM	HDD
bdlogin01.uo.hpc.cu	8	64GB	400GB
bdlogin02.uo.hpc.cu	12	32GB	400GB

- Nodos de cálculo

Nombre de dominio	Cores	RAM	HDD
bdcalc01.uo.hpc.cu	64	64GB	600GB
bdcalc02.uo.hpc.cu	64	64GB	600GB
bdcalc03.uo.hpc.cu	64	64GB	600GB

# RECURSOS DEL CLÚESTER

## Hardware:

- Nodos de gestión.

Nombre de dominio	Cores	RAM	HDD
bdmaster01.uo.hpc.cu	12	64GB	400GB
bdmaster02.uo.hpc.cu	12	64GB	400GB

# RECURSOS DEL CLÚESTER

## Software (HDP 3.1):

Software	Version	Description
HDFS	3.1.1.3.1	Apache Hadoop <b>Distributed File System</b>
YARN	3.1.0	Framework for task planning and cluster resource management.
MapReduce2	3.0.0.3.1	MapReduce model implementation based on YARN
Tez	0.9.0.3.1	Tez is the next generation <b>Hadoop Query Processing framework</b> written on top of YARN.
Hive	3.0.0.3.1	<b>Data warehouse system</b> for ad-hoc queries & analysis of large datasets and table & storage management service
Hbase	2.0.0.3.1	<b>Non-relational distributed database</b> and centralized service for configuration management & synchronization

# RECURSOS DEL CLÚESTER

## Software (HDP 3.1):

Software	Version	Description
Pig	0.16.1.3.1	Scripting platform for analyzing large datasets
Sqoop	1.4.7	Tool for transferring bulk data between Apache Hadoop and structured data stores such as relational databases
Storm	1.2.1	Apache Hadoop <b>Stream processing framework</b>
Accumulo	1.7.0	Robust, scalable, high performance <b>distributed key/value store.</b>
Kafka	1.0.0.3.1	A high-throughput <b>distributed messaging system</b>
Spark2	2.3.0	Apache Spark 2.3 is a fast and <b>general engine for large-scale data processing.</b>

## Acceso y almacenamiento temporal:

- Cuenta de acceso al clúster.
  - Solicitud a través de <https://portal.uo.hpc.cu/website/>
- Acceso a los comandos de trabajo con el clúster.
  - spark-shell, pyspark, spark-submit, hdfs, ...
- Carpeta personal en cada nodo de acceso.
  - /home/CLUSTER/nombredeusuario/...
- Carpeta personal en HDFS.
  - hdfs:///user/nombredeusuario/...

# RECURSOS DEL USUARIO

## Herramientas de apoyo:

- Zeppelin Notebook 0.8.0.
  - <http://bdlogin01.uo.hpc.cu:9995/>
  - <http://bdlogin02.uo.hpc.cu:9995/>
- Interface del planificador.
  - <http://bdmaster01.uo.hpc.cu:8088/ui2>
  - <http://bdlogin01.uo.hpc.cu:18081/> (monitoreo)
- Interface de información de HDFS.
  - <http://bdmaster01.uo.hpc.cu:50070/>
- Explorador HDFS.
  - [http://bdmaster01.uo.hpc.cu:50070/explorer.html#/.](http://bdmaster01.uo.hpc.cu:50070/explorer.html#/)

# APLICACIONES SPARK

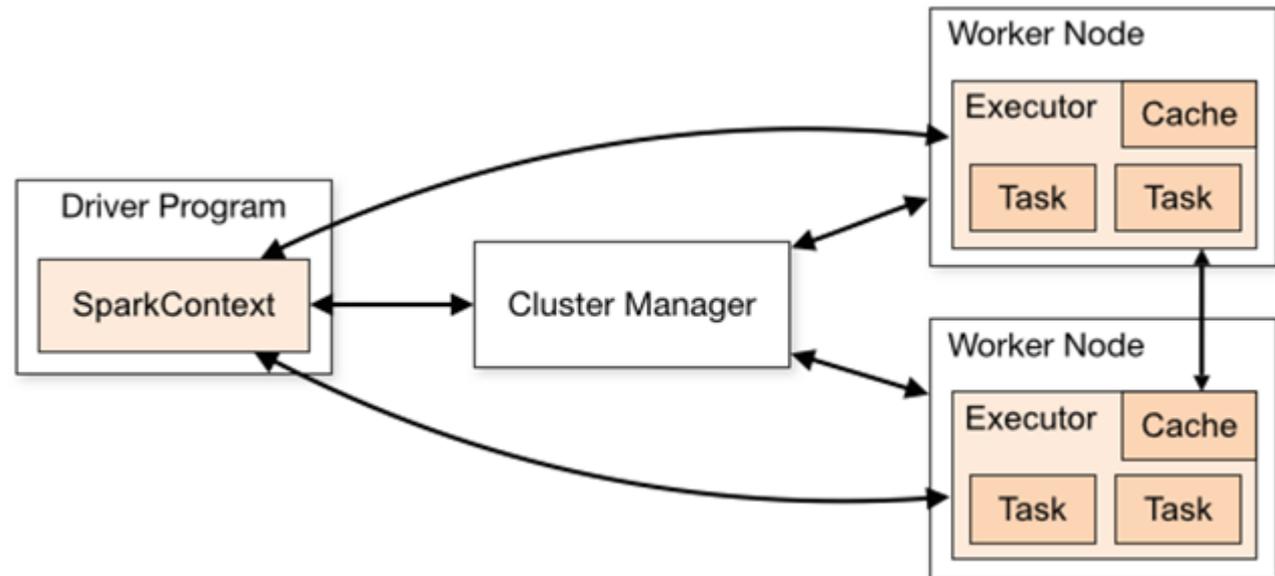
## Visión general desde un clúster:

- Corren como un conjunto de procesos independientes en el clúster, coordinados por el objeto *SparkContext* del programa principal (driver program).
- SparkContext puede conectarse a varios tipos de cluster managers.
  - Spark Standalone, Mesos o YARN.
- Una vez conectado, adquiere ejecutores en los nodos del clúster.
  - Son procesos que ejecutan cálculos y almacenan datos para la aplicación.

# APLICACIONES SPARK

## Visión general desde un clúster: :

- Luego, envía el código de la aplicación a los ejecutores.
  - Archivos JAR o Python pasados a SparkContext.
- Finalmente, SparkContext envía las tareas a los ejecutores.



# EJECUCIÓN DE APLICACIONES

## Corriendo una aplicación Spark:

- Para ejecutar aplicaciones Spark en un clúster se utiliza el comando `spark-submit`.
- El comando está disponible en los nodos de acceso.
  - SSH -> `bdlogin01.uo.hpc.cu / bdlogin02.uo.hpc.cu`
- Por tanto es necesario colocar las aplicaciones en algún nodo de acceso para luego someterlas a ejecución en el clúster.
  - Ej.: `/home/CLUSTER/uo_oscar/myprograma.jar`
  - WinSCP.exe y el comando Linux “cp” facilitan este paso.

# EJECUCIÓN DE APLICACIONES

## Corriendo una aplicación Spark en YARN:

- Opciones más comunes del comando `spark-submit`.

`$spark-submit \`

`--class <main-class> \` -> Indica el punto de entrada de la aplicación.

`--master <master-url> \` -> Resource manager (“yarn” en nuestro caso).

`--deploy-mode <deploy-mode> \` -> “client” o “cluster”

`--conf <key>=<value> \` -> Propiedades de configuración de Spark.

`... # otras opciones`

`<application-jar> \` -> Ruta del archivo .jar (aplicación).

`[application-arguments]` -> Argumentos de la función `main()`.

# EJECUCIÓN DE APLICACIONES

## Corriendo una aplicación Spark en YARN:

- Ejemplo de una ejecución.

```
$spark-submit \  
  --class org.apache.spark.examples.SparkPi \  
  --master yarn \  
  --deploy-mode client \  
  /usr/hdp/3.1.0.0-78/spark2/examples/jars/spark-examples.jar \  
  1000
```



# Introducción al uso del clúster de Big Data de la UO.

**Autor: Robert Amaurys Reina Valladares.**

Dirección de Informatización, Universidad de Oriente (UO).  
Departamento de Informática, Facultad de Ingeniería en Telecomunicaciones, Informática y Biomédica, UO.  
Mar. 2020 / Santiago de Cuba.