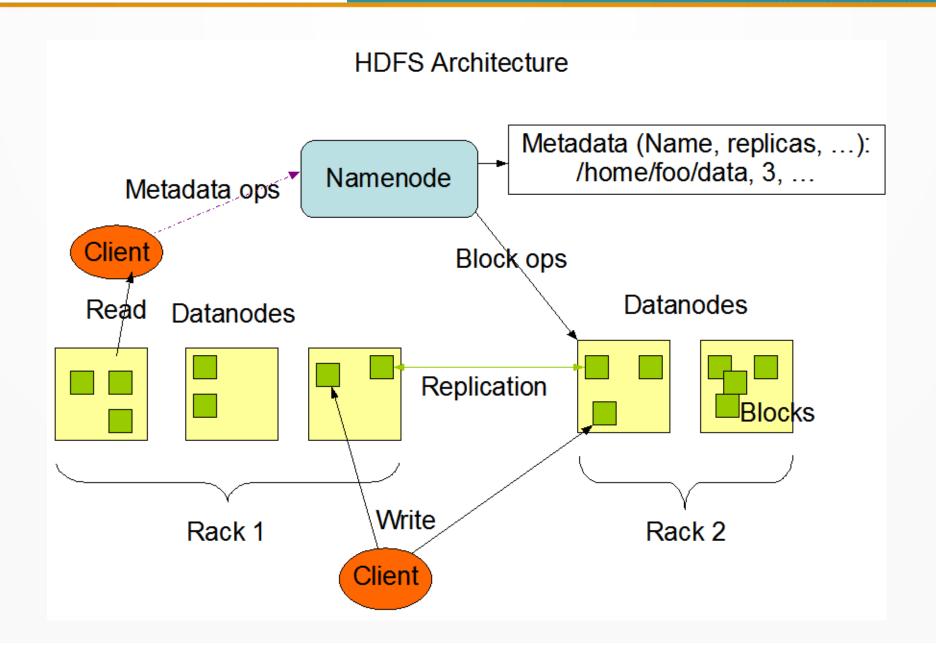
# **CURSO DE PROGRAMACION SCALA**Sesión 14

Sergio Couto Catoira

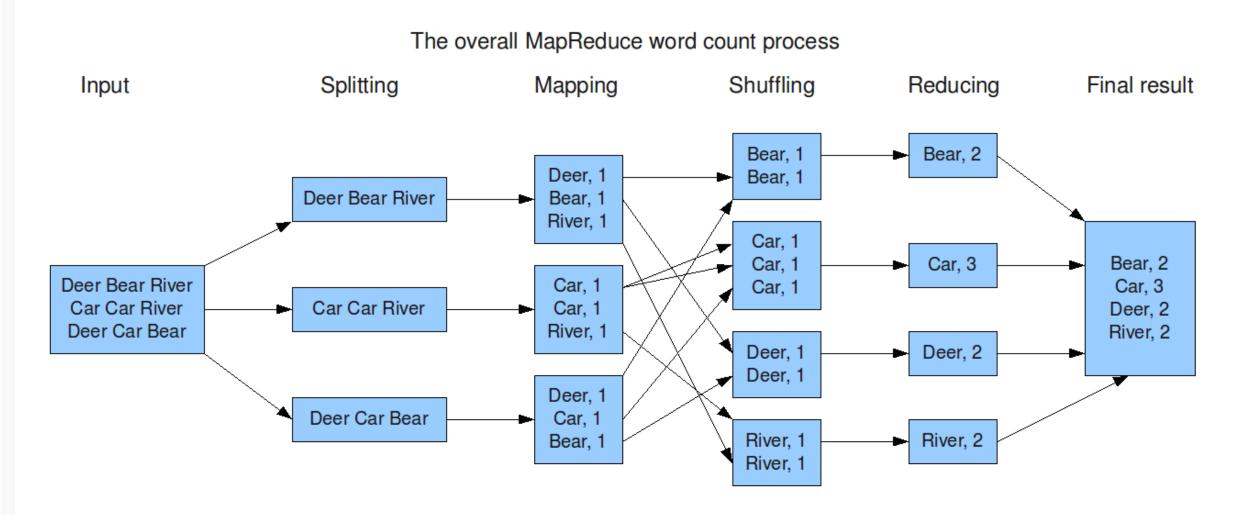
## Índice

- >Hadoop
- > HDFS
- Spark (http://spark.apache.org/downloads.html)
  - Arquitectura
  - APIs
  - Otros Conceptos
- > Ejercicio sobre RDD
- > Ejercicio sobre Dataframe
  - UDF



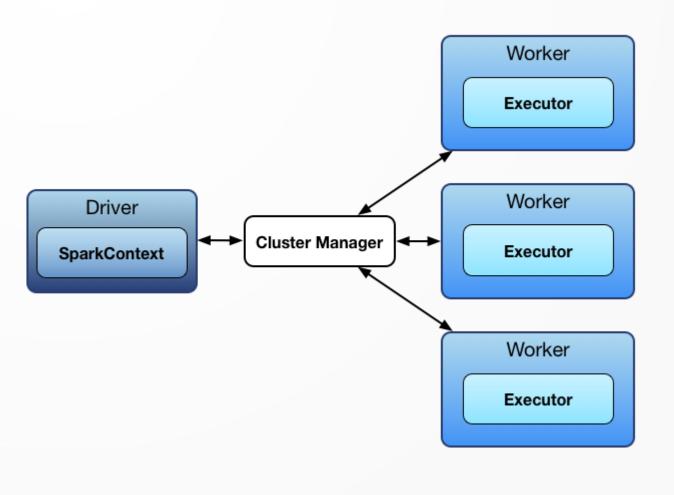


# MapReduce



#### Spark - Arquitectura

- Driver: JVM propia
- > Worker: Instancias
  - Cada uno al menos un ejecutor
- Executor: JVM Propia(a partir de Spark 2 )
  - Ejecuta las tareas con los datos de su nodo
- Task: Cada executor puede



# Spark - API

- >RDD (Resilient distributed dataset)
  - Sin esquema
- DataFrame
  - Con esquema (similar a una tabla)
  - Datos organizados en columnas con nombre
- Dataset (a partir de Spark 2 )
  - Unifica API con Dataframe

### Spark - Conceptos

- >Laziness
  - Transformaciones
  - Acciones
- Shuffle and Sort
- >Spill a disco
- >Spark-shell

#### RDD - Ejercicios

- >WordCount Genera un programa que cuenta las palabras del fichero y las genere en un fichero de salida ordenado (src/main/resources/Shakespeare.txt)
- Los pasos a ejecutar:
  - Leer fichero (sc.TextFile)
  - Splitearlo en palabras
  - Eliminar símbolos y pasar todo a mayusculas/minusculas (.replaceAll("[,.!?:;\"')(-]", ""))
  - Filtrar espacios
  - Contar palabras
  - Ordenar
  - Escribir fichero de salida

#### RDD - Ejercicios

- Alturas Genera un programa que lea el fichero alturas.csv y calcula la media de altura por sexo
- Debes filtrar datos erróneos (vacíos o negativos) y corregir los que vengan mal (en metros en lugar de en centímetros)
- Métodos a tener en cuenta
  - groupByKey => Luego lo sustituiremos por aggregate
  - Map
  - Filter
  - toDouble

# Dataframe - Ejercicios

- Alturas Mismo programa que el anterior pero con dataframes
- > Métodos a tener en cuenta
  - Leer df => ss.read.csv("src/main/resources/alturas.csv")
  - ToDF => Permite pasar un esquema
  - WithColumn => Crea una columna
  - myDataFrame("columna") => Select de una columna