



CTIC UNI

Centro de Tecnologías de Información y Comunicaciones
Universidad Nacional de Ingeniería

Tecnologías para el Big Data

Agenda



CTIC UNI

Centro de Tecnologías de Información y Comunicaciones
Universidad Nacional de Ingeniería

- ^ Introducción
- ^ Tecnologías batch procesamiento
- ^ Tecnologías batch analítica
- ^ Tecnologías real time/near real time
- ^ Ejercicios Prácticos

Compartamos



CTIC UNI

Centro de Tecnologías de Información y Comunicaciones
Universidad Nacional de Ingeniería

Coméntanos sobre el artículo que leíste

Introducción

Objetivos

- Entender el nuevo paradigma del Big Data.
- Comprender las necesidades de Sistemas Batch
- Comprender las necesidades de Sistemas Online



CTIC UNI

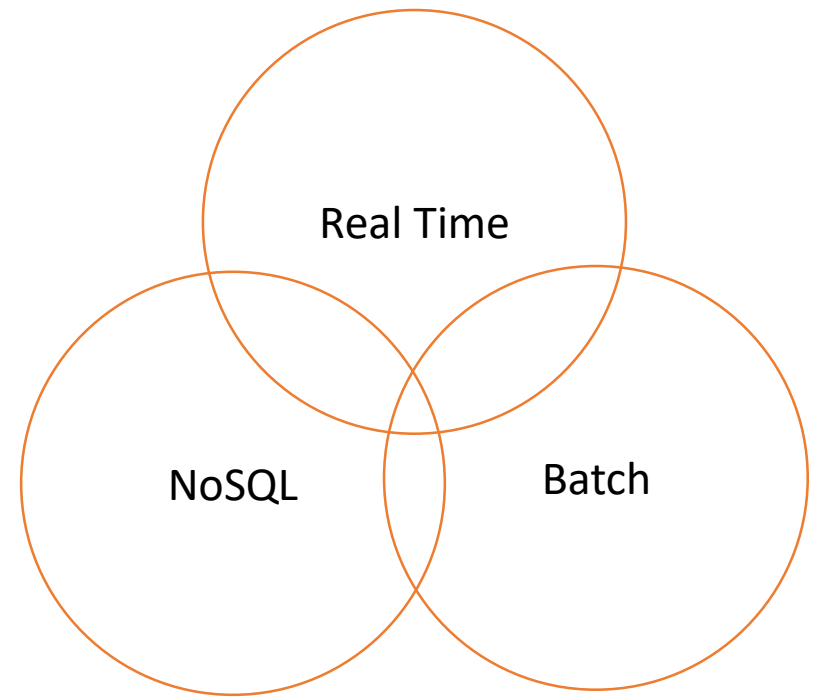
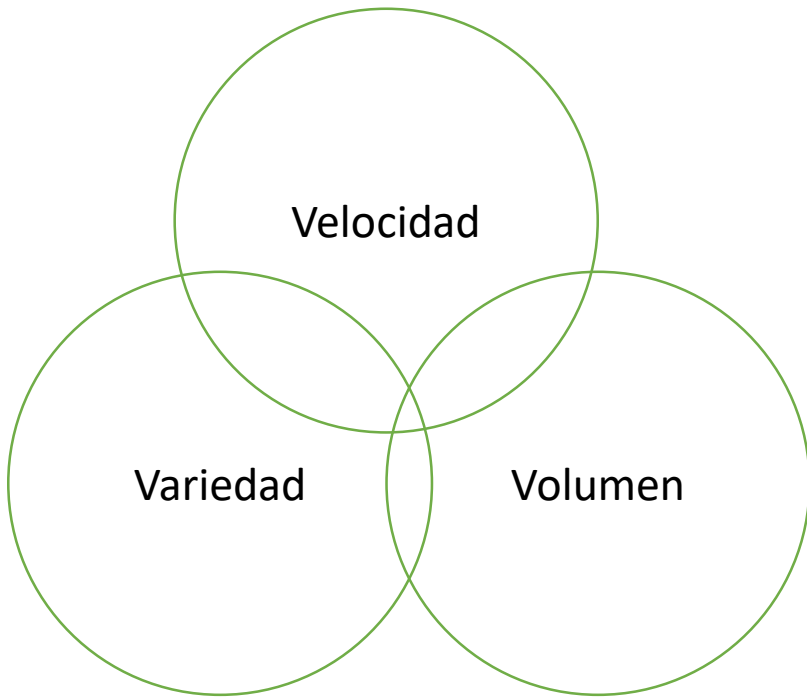
Centro de Tecnologías de Información y Comunicaciones
Universidad Nacional de Ingeniería

Introducción



CTIC UNI

Centro de Tecnologías de Información y Comunicaciones
Universidad Nacional de Ingeniería



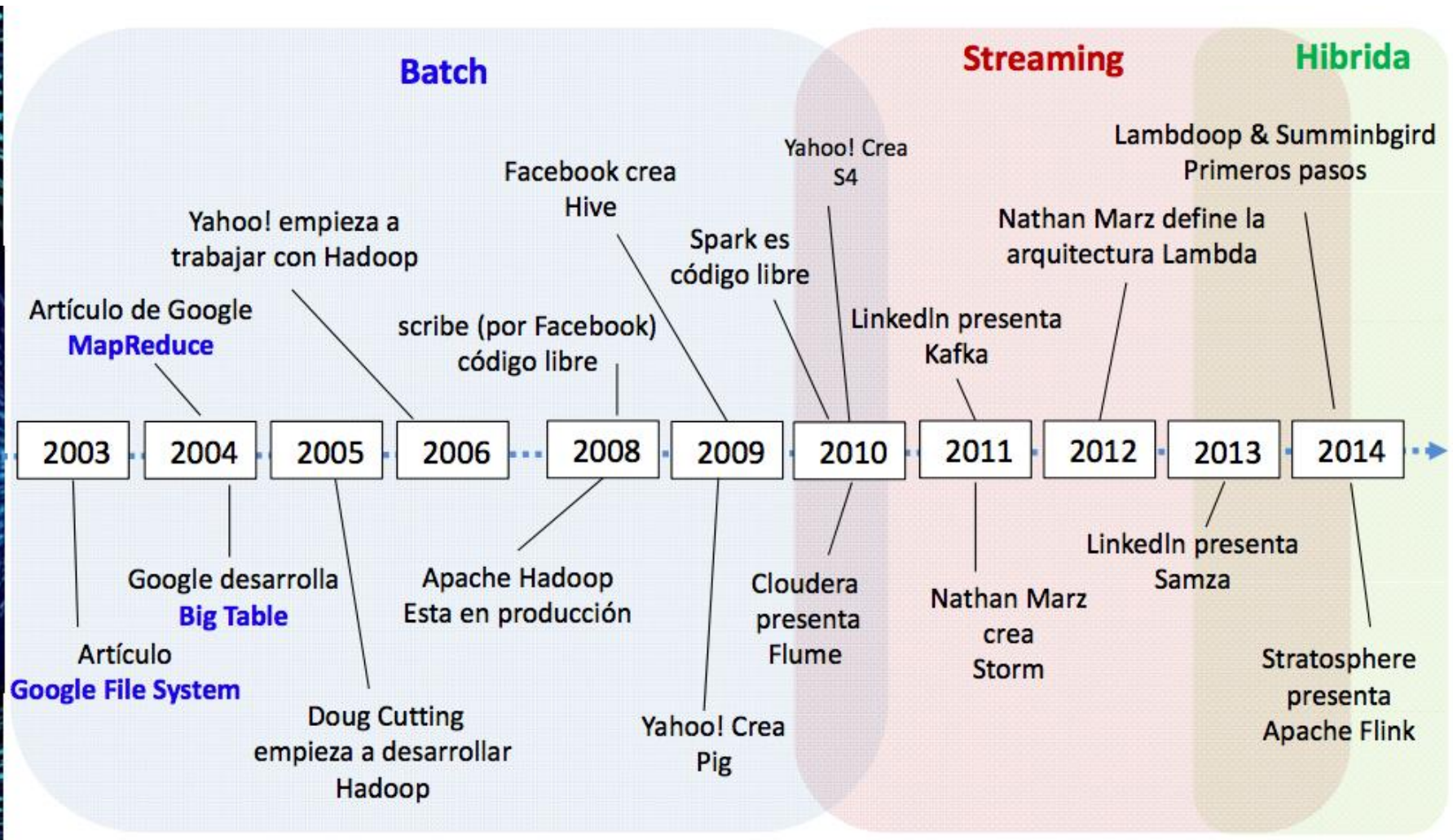
Grandes problemas, grandes soluciones

Introducción



CTIC UNI

Centro de Tecnologías de Información y Comunicaciones
Universidad Nacional de Ingeniería



[ECV, 2016]

Tecnologías batch procesamiento



Objetivos

- Entender el nuevo paradigma del Big Data.
- Solución para el problema del volumen
- Procesamiento grandes cantidad de datos
- Escalable
- Procesamiento distribuido y paralelo
- Tolerancia a fallos
- Alta latencia

Tecnologías batch



CTIC UNI

Centro de Tecnologías de Información y Comunicaciones
Universidad Nacional de Ingeniería

Hadoop

- Principal herramienta para la implementación de una arquitectura batch processing.
- Se han ido desarrollando nuevas herramientas para cada una de las necesidades

Comandos HDFS

HDFS

Map Reduce

Adquisición

Almacenamiento

Análisis

Presentación

¿Qué es Hadoop?

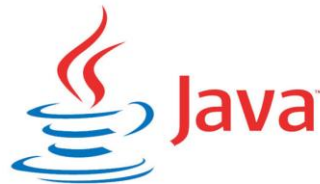


CTIC UNI

Centro de Tecnologías de Información y Comunicaciones
Universidad Nacional de Ingeniería

Hadoop

- Librería que permite manipular grandes volúmenes de información en forma distribuida.
 - Creado en 2006 por Doug Cutting inspirado en artículos Google file system y Map Reduces
- Diseñada para escalar de unos pocos a hasta miles de servidores
- Formado: HDFS para almacenamiento y Map Reduce para procesamiento.



¿Qué es Hadoop?



HDFS

- Sistema de ficheros distribuido en Hadoop
- Puede ser usado en Commodity hardware
- Almacenar grandes volúmenes de datos
- Procesar mucha información utilizando MapReduce.
- Elevado ancho de banda
- Tolerancia a fallos / replicación
- “Write once, read more”

¿Qué es Hadoop?



NameNode

- Realiza la gestión del clúster
- Almacena y gestiona los metadatos de los ficheros y directorios
- La información de los metadatos se almacena en memoria RAM
- Regula acceso a los ficheros por parte de los clientes
- Necesita más potencia que los otros nodos

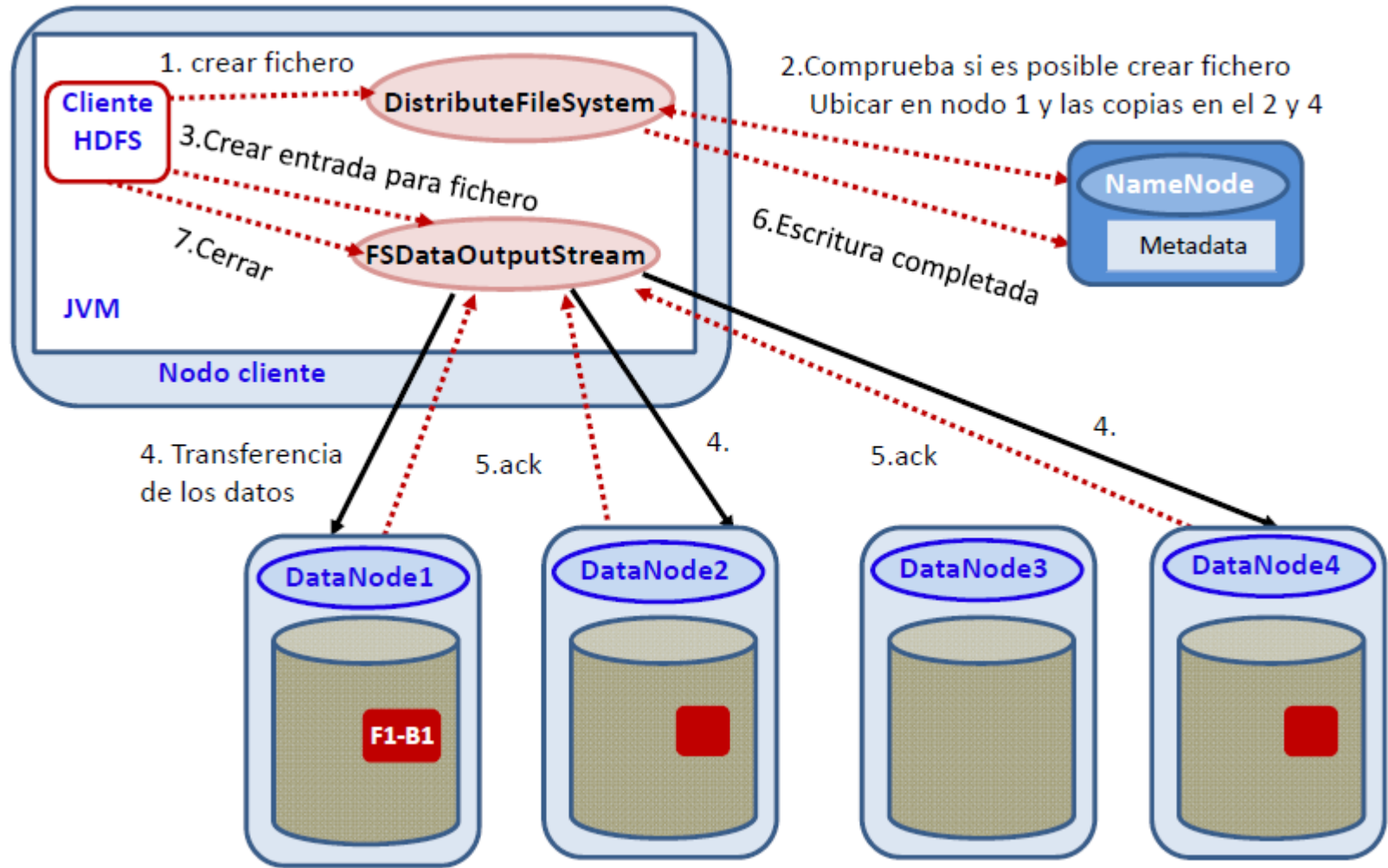
¿Qué es Hadoop?



DataNode

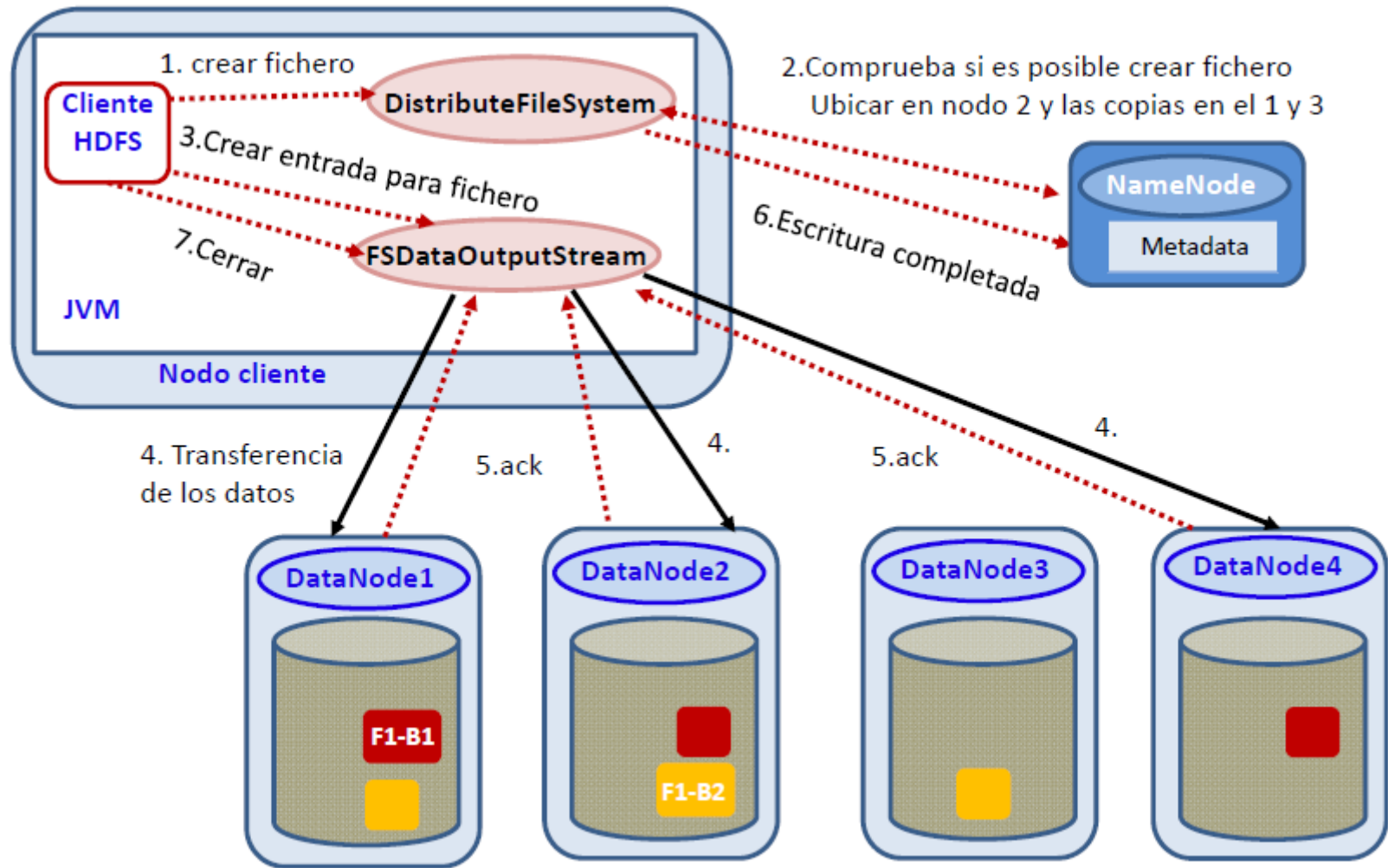
- Almacenan los datos (bloques ficheros)
- Responsable de leer y escribir las peticiones de los clientes
- Informan al DataNode de los bloques almacenados

HDFS-Escritura



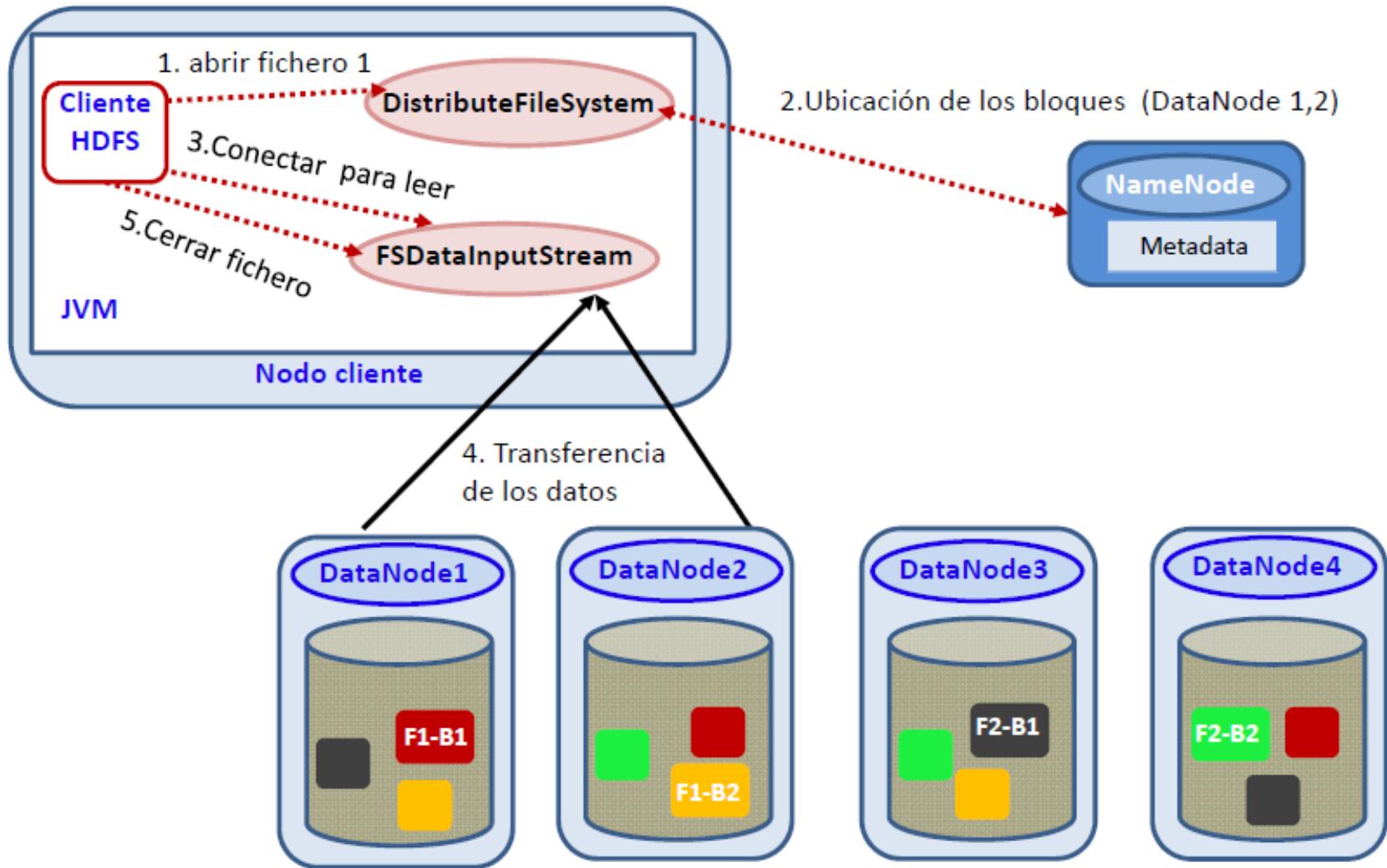
[ECV, 2016]

HDFS-Escritura



[ECV, 2016]

HDFS-Lectura



Interactuar HDFS



CTIC UNI

Centro de Tecnologías de Información y Comunicaciones
Universidad Nacional de Ingeniería

Comando	Resultado
hdfs dfs -ls <path> hdfs dfs -ls -R <path>	Lista ficheros Lista recursivamente
hadoop fs -cp <src> <dst> hadoop fs -mv <src> <dst>	Copia ficheros HDFS a HDFS Mueve ficheros HDFS a HDF
hadoop fs -rm <path> hdfs dfs -rm -r <path>	Borra ficheros en HDFS Borra recursivamente
hdfs dfs -cat <path> hdfs dfs -tail <path> hdfs dfs -stat <path>	Muestra fichero en HDFS Muestra el final del fichero Muestra estadísticas del fichero
hdfs dfs -mkdir <path>	Crea directorio en HDFS
hdfs dfs -chmod ... hdfs dfs -chown ...	Cambia permisos de fichero Cambia propietario/grupo de fichero
hdfs dfs -put <local> <dst> hdfs dfs -copyFromLocal ... hdfs dfs -moveFromLocal ...	Copia de local a HDFS Igual que -put Mueve de local a HDFS
hdfs dfs -get <src> <loc> hdfs dfs -copyToLocal ... hdfs dfs -getmerge ...	Copia de HDFS a local Copia de HDFS a local Copia mezclando y ordenando
hdfs dfs -text <path>	Muestra el fichero en texto

Interactuar HDFS



Adicional:

`hdfs dfsadmin` – Permite obtener información del estado del HDFS.

`hdfs fsck` – Revisar la salud de los fichero en HDFS

`sbin/start-balancer.sh` – Redistribuir bloques entre los datanodes

MapReduce



- Modelo de programación para el procesamiento paralelo de grandes conjuntos de datos (Diseñado por google, 2004).
- Popularizado por Apache Hadoop
- Basado programación funcional y algoritmo divide y vencerás
- Solo dos funciones map y reduce
- Programas automáticamente paralelizados

MapReduce



Divide y Vencerás

- Descomponer un problema en subproblemas (más pequeños)
- Resolver cada subproblema
- Combinar soluciones

Pseudocódigo

Función DV (X: Datos del problema)

si pequeño(X) entonces

 S = sencillo(X)

sino

 descomponer(X) => (X1, ..., Xk)

 para i = 1 hasta k hacer

 Si = DV(Xi)

 S = combinar(S1, ..., Sk)

devolver S

MapReduce

Divide y Vencerás

- Debe elegirse el umbral para aplicar $\text{sencillo}(x)$
- Funciones de descomposición y combinación deben ser eficientes
- Subproblemas del mismo tamaño



MapReduce



Función map

- Para cada par de entrada clave/valor genera una lista de pares claves/valores intermedios

$$\text{map}(K_1, V_1) \rightarrow \text{list}(K_2, V_2)$$

Función reduce

- Combina valores intermedios para cada clave particular

$$\text{reduce}(K_2, \text{list}(V_2)) \rightarrow (K_3, \text{list}(V_3))$$

MapReduce



CTIC UNI

Centro de Tecnologías de Información y Comunicaciones
Universidad Nacional de Ingeniería

Fase Input

- Divide los datos
- Prepara los datos → claves/valor

Fase Map

- Ejecuta la función map para cada par clave/valor

Fases Shuffle & Sort

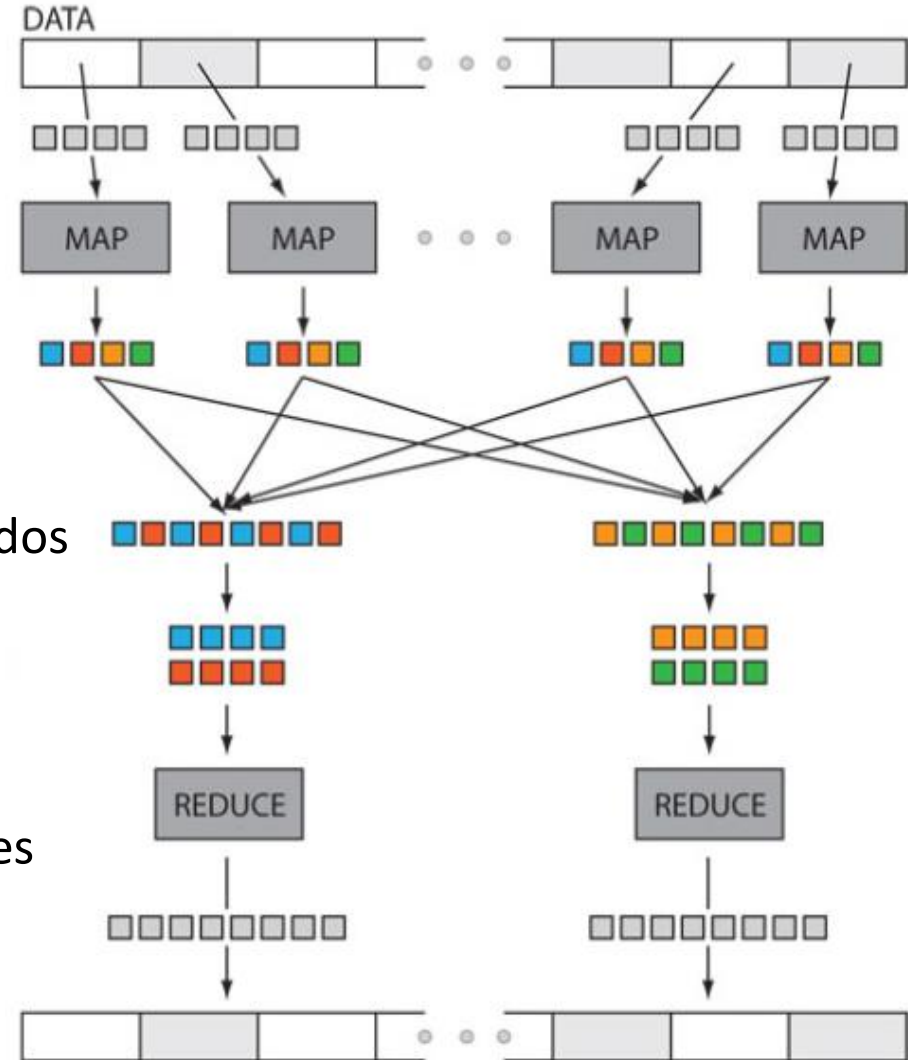
- Ordena y agrupa por clave los resultados de la fase map

Fase Reduce

- Ejecuta la función reduce
- Puede haber una o múltiples funciones Reduce

Fase Output

- Almacena el/los resultados en un sistema de ficheros , una base de datos





Ejemplo Map Reduce

1-Contar el número de veces que aparece cada palabra en un fichero de texto

Entrada:

- *Clave = podría ser el nombre del fichero o el número de línea pero en este caso **no se necesita***
- *Valor = línea de texto*

Salida:

- *pares (palabra, n° de ocurrencias)*
- Pseudocódigo del Map
- *// key: nada, value: línea de texto*

Map (key, value)

for each word **w** in value **emit(w, 1)**

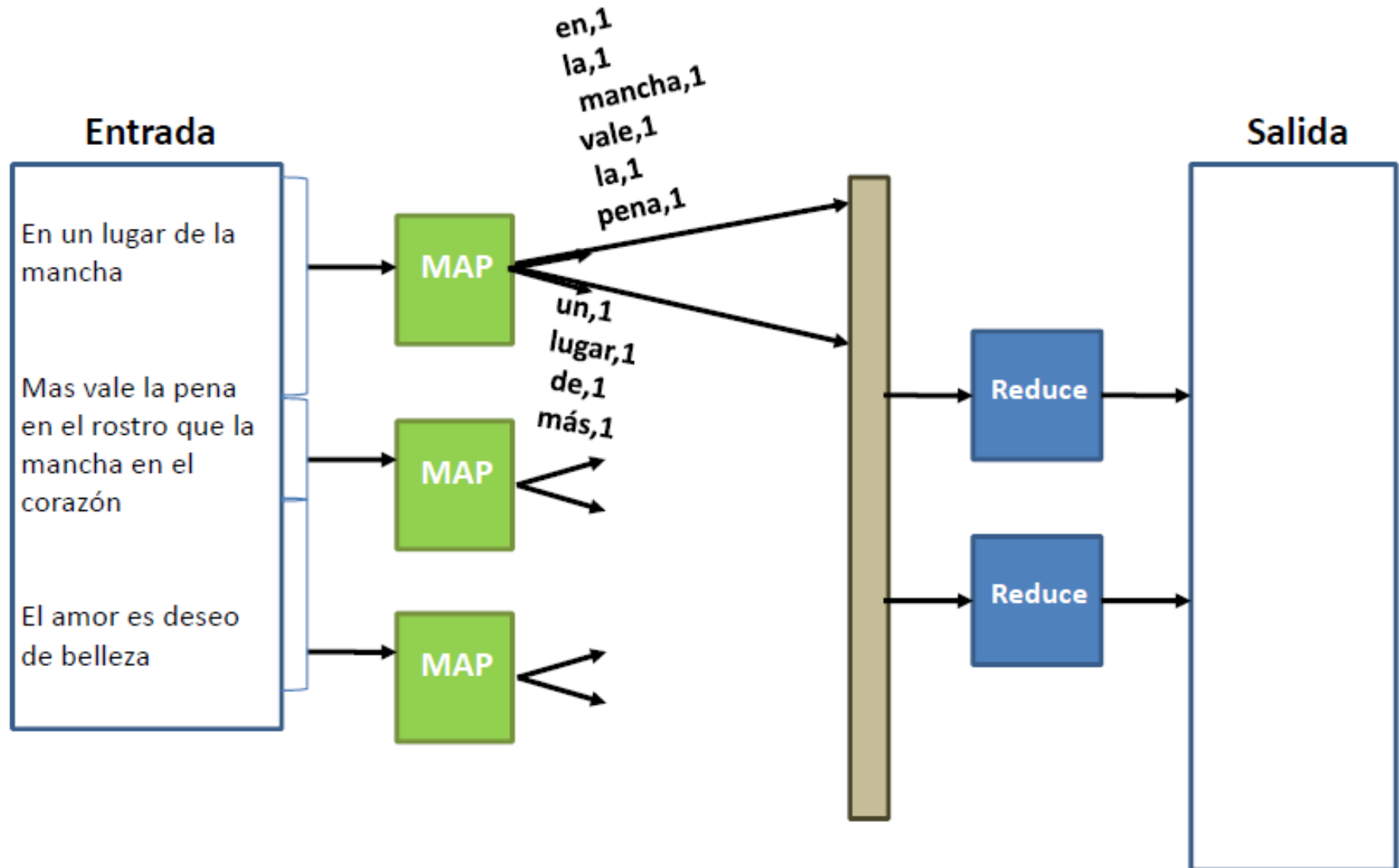
- Pseudocódigo del Reduce

// Key: palabra; values: un iterador sobre los 1s

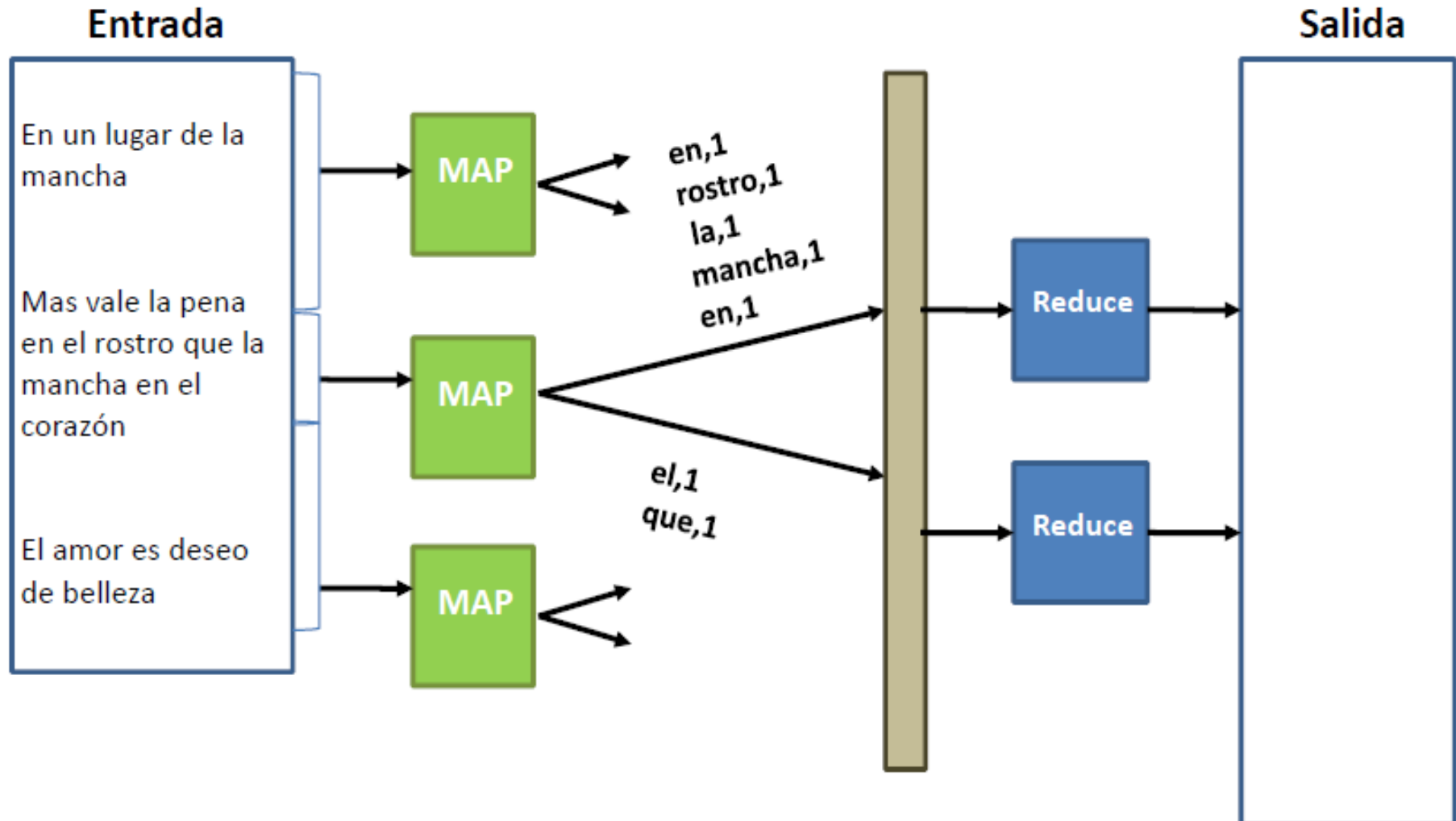
Reduce(key, values)

emit(key, sum(values))

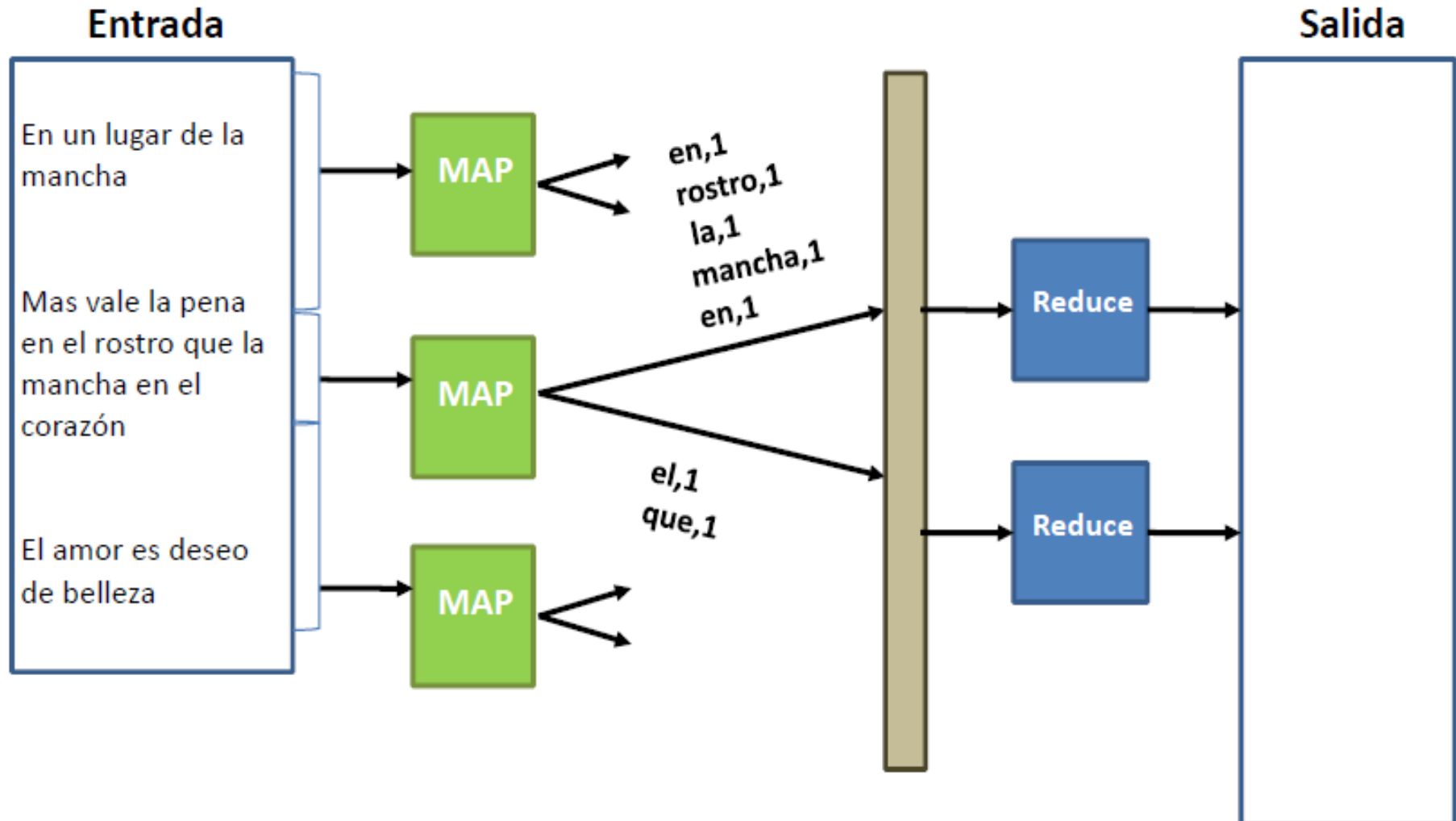
Ejemplo Map Reduce



Ejemplo Map Reduce



Ejemplo Map Reduce

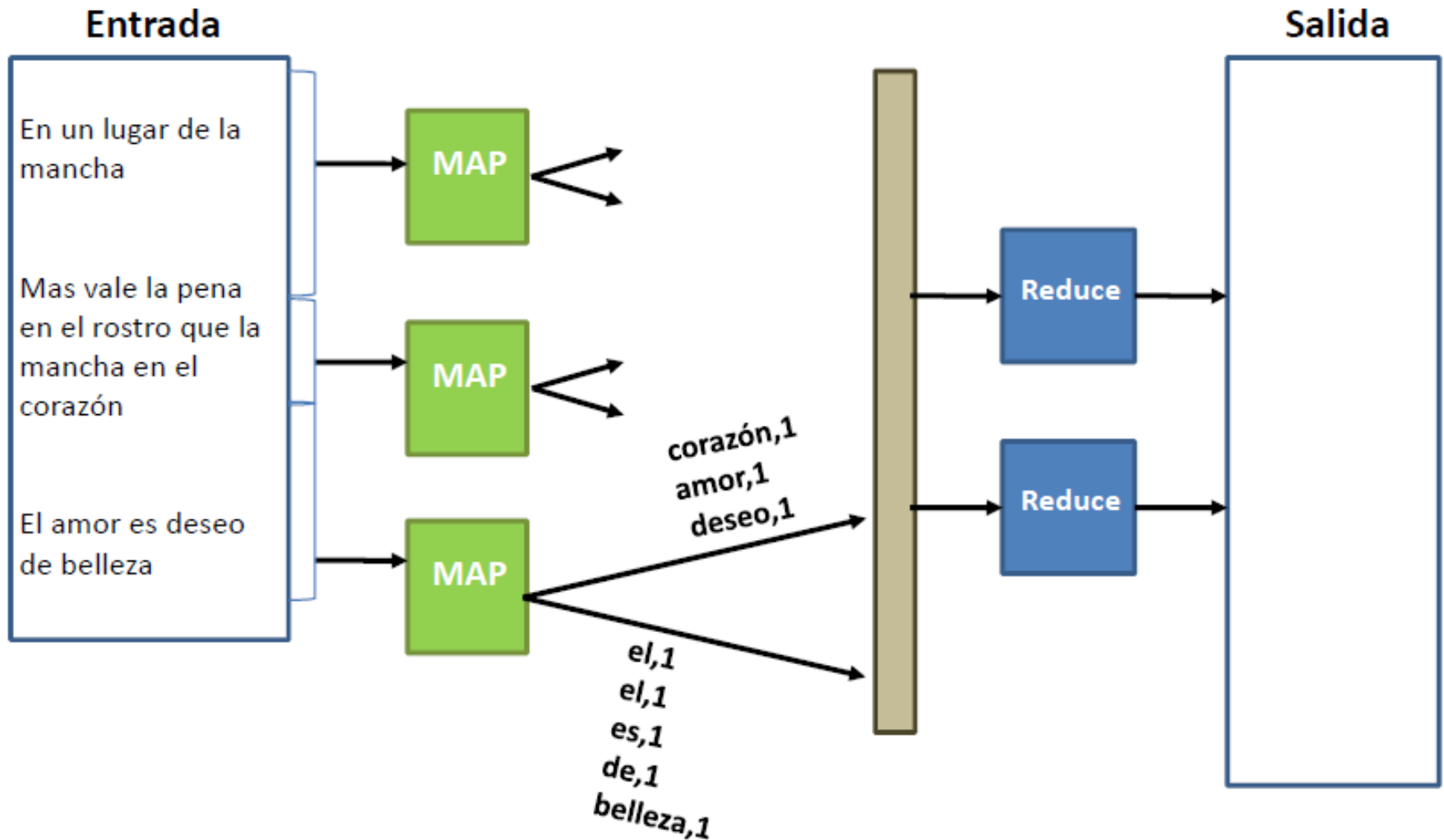


Interactuar HDFS



CTIC UNI

Centro de Tecnologías de Información y Comunicaciones
Universidad Nacional de Ingeniería

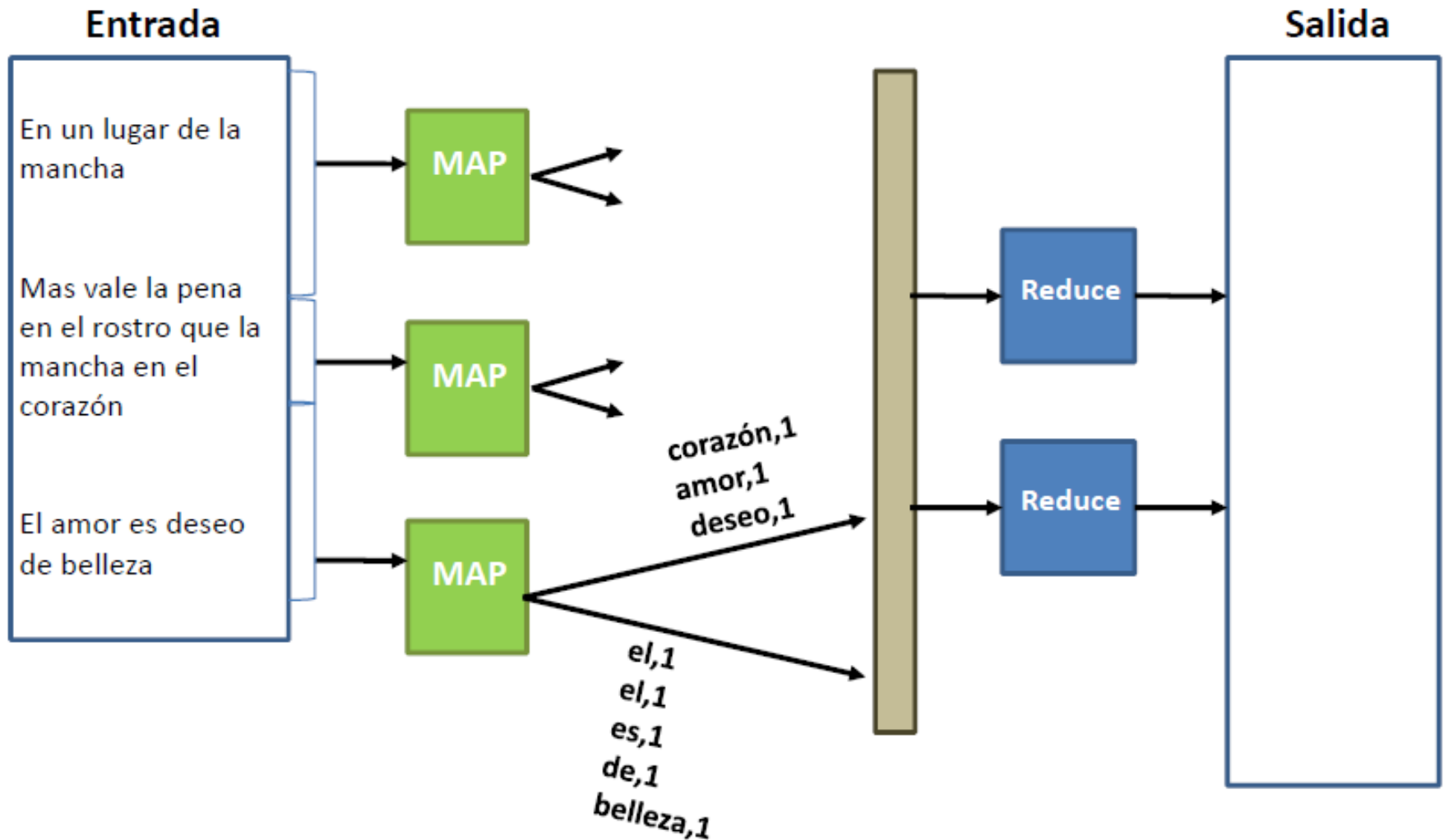


Interactuar HDFS

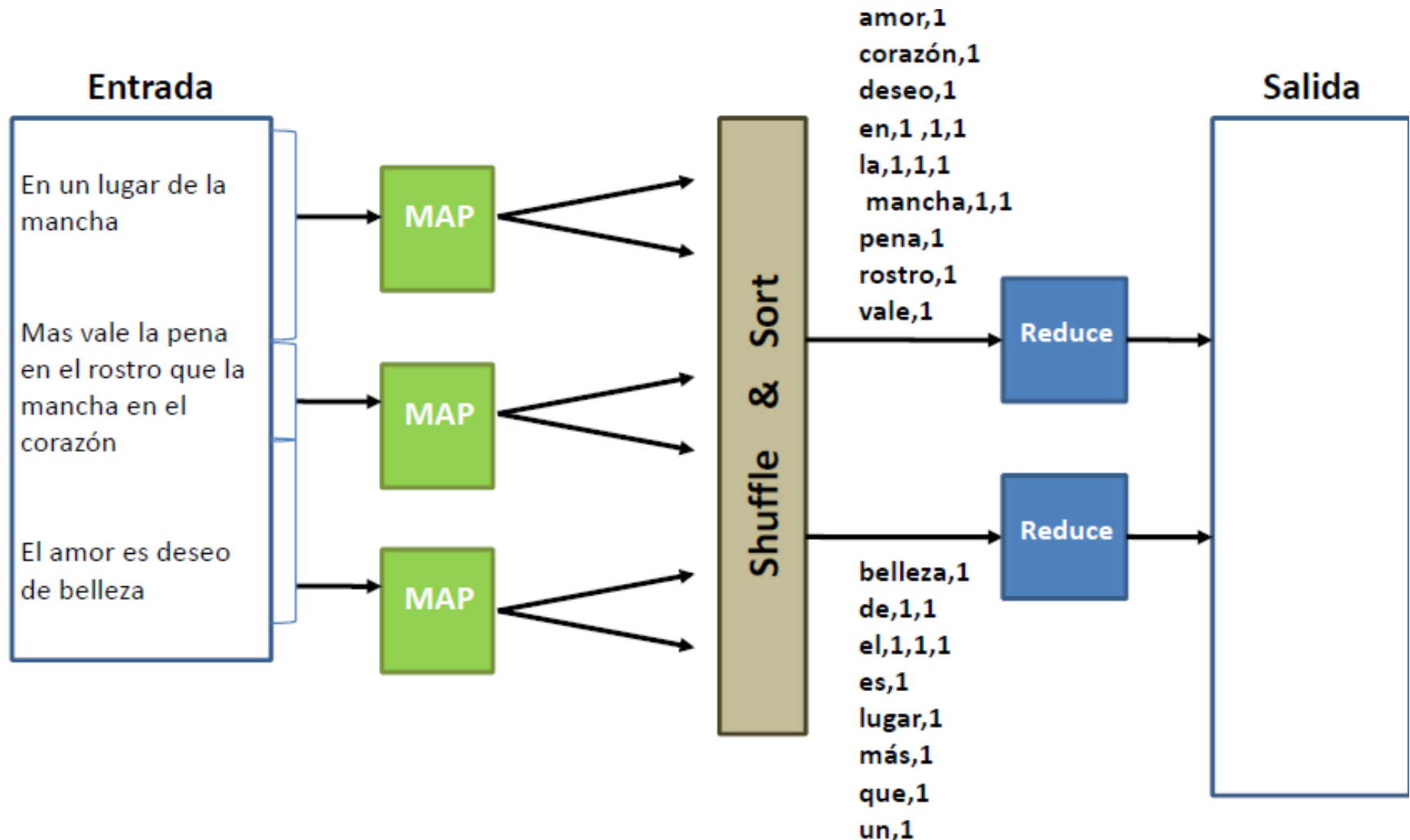


CTIC UNI

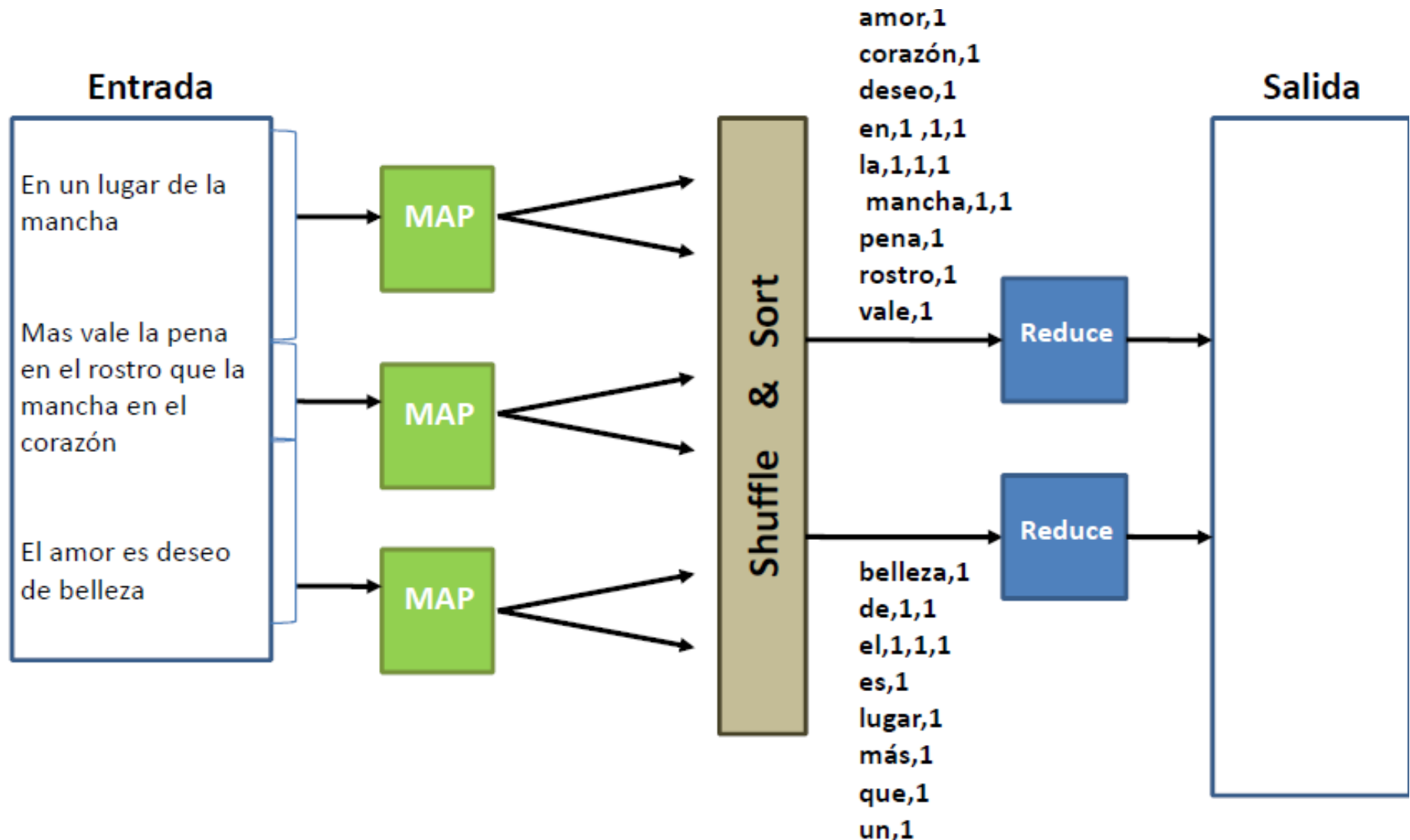
Centro de Tecnologías de Información y Comunicaciones
Universidad Nacional de Ingeniería



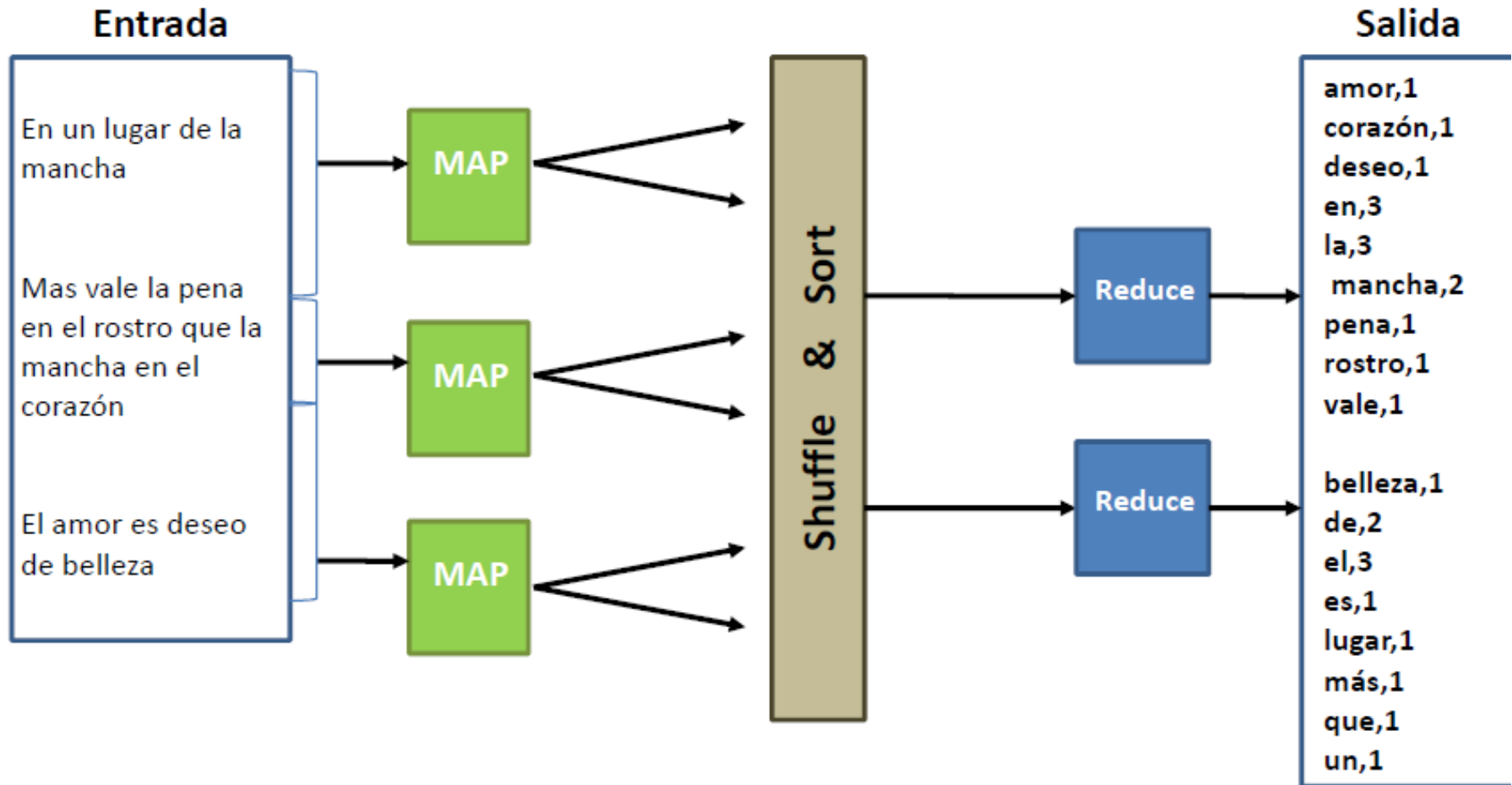
Interactuar HDFS



Interactuar HDFS



Interactuar HDFS



Interactuar HDFS



CTIC UNI

Centro de Tecnologías de Información y Comunicaciones
Universidad Nacional de Ingeniería

Ahora en equipos

En cuántos ficheros está cada palabra

Clave = ???

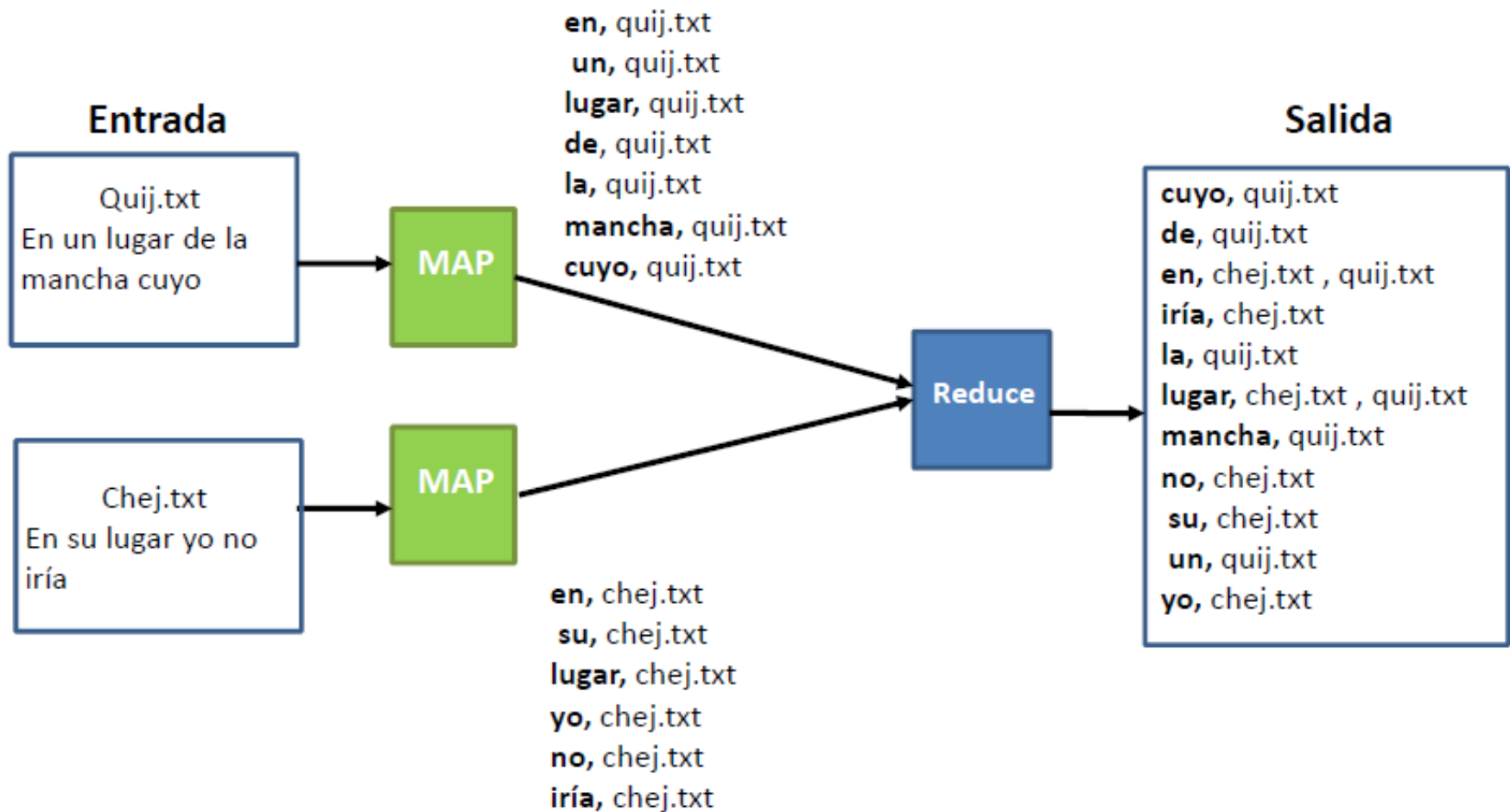
Valor = ???

Entrada

Quij.txt
En un lugar de la
mancha cuyo

Chej.txt
En su lugar yo no
iría

Interactuar HDFS



Modelo de ejecución

Map Reduce



- Minimizar el movimiento de datos
- Las funciones map y reduce se ejecutan en los nodos donde se encuentra los nodos.
- Facilita la distribución del trabajo
- El programador solo se encarga de programar las funciones map() y reduce()
- Tolerante a fallos
- Altamente escalable

Modelo de ejecución

Map Reduce



Map Worker

- Se inician en el mismo nodo que sus datos de entrada
- Graban su salida en el disco local antes de pasarla a los reduce worker
- Permite recuperar reduce en caso de fallos

Reduce Worker

- Graban su salida en el sistema distribuido
- Un fichero de salida por cada reduce

Modelo de ejecución Map Reduce



Master

- Estado de cada tarea: *en-reposo* , *en-progreso*, *completada*
- Las tareas *en-reposo* se planifican a medida que van quedando workers libres
- Cuando una tarea map acaba, envía al Master la localización y tamaño de sus ficheros intermedios (uno por cada tarea reduce)
- El Master envía esta información a las tareas reduce, cuando acaban todos los map

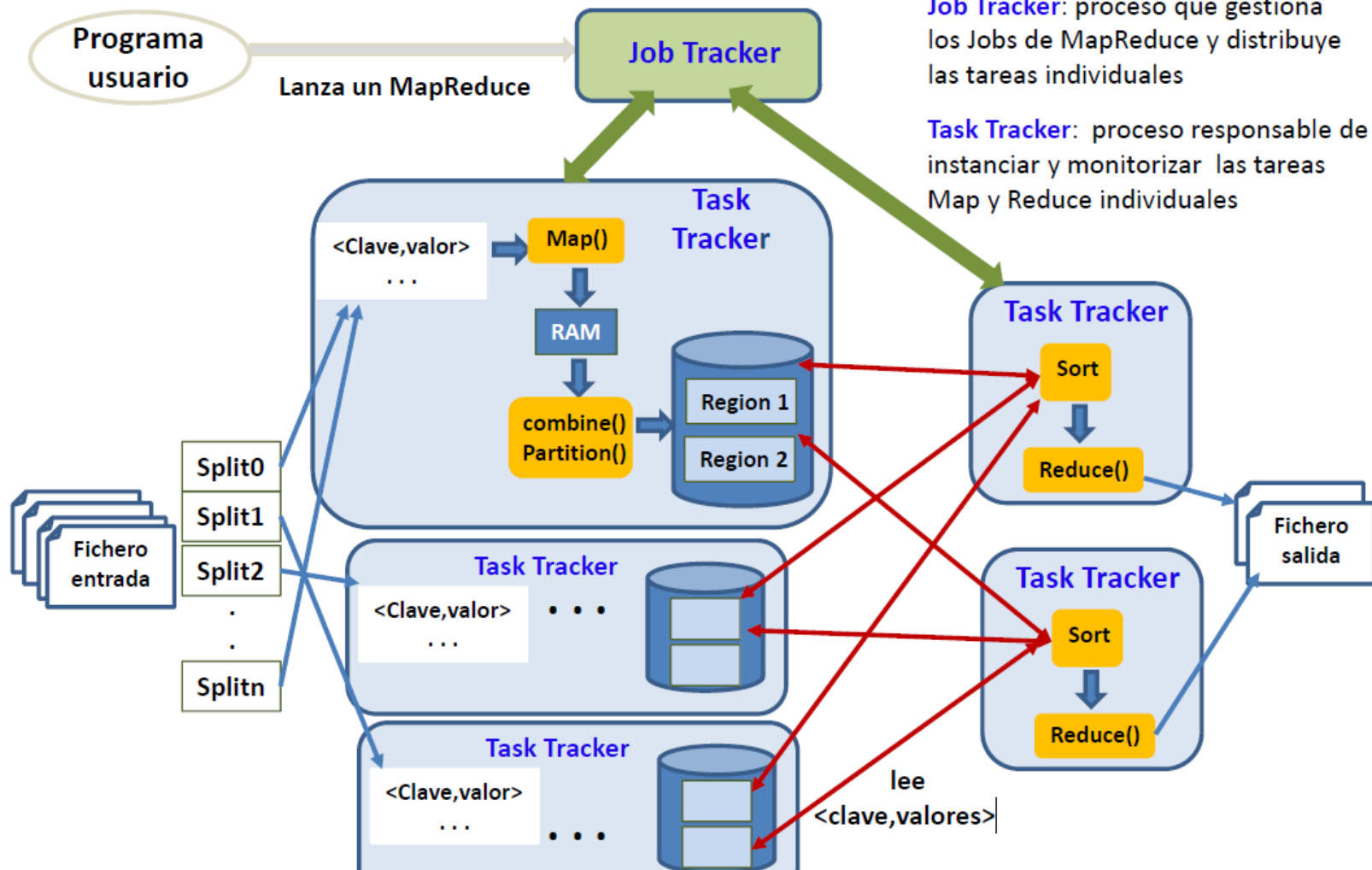
Modelo de ejecución Map Reduce



Master

- Estado de cada tarea: *en-reposo* , *en-progreso*, *completada*
- Las tareas *en-reposo* se planifican a medida que van quedando workers libres
- Cuando una tarea map acaba, envía al Master la localización y tamaño de sus ficheros intermedios (uno por cada tarea reduce)
- El Master envía esta información a las tareas reduce, cuando acaban todos los map

Modelo de ejecución Map Reduce



Modelo de ejecución Map Reduce

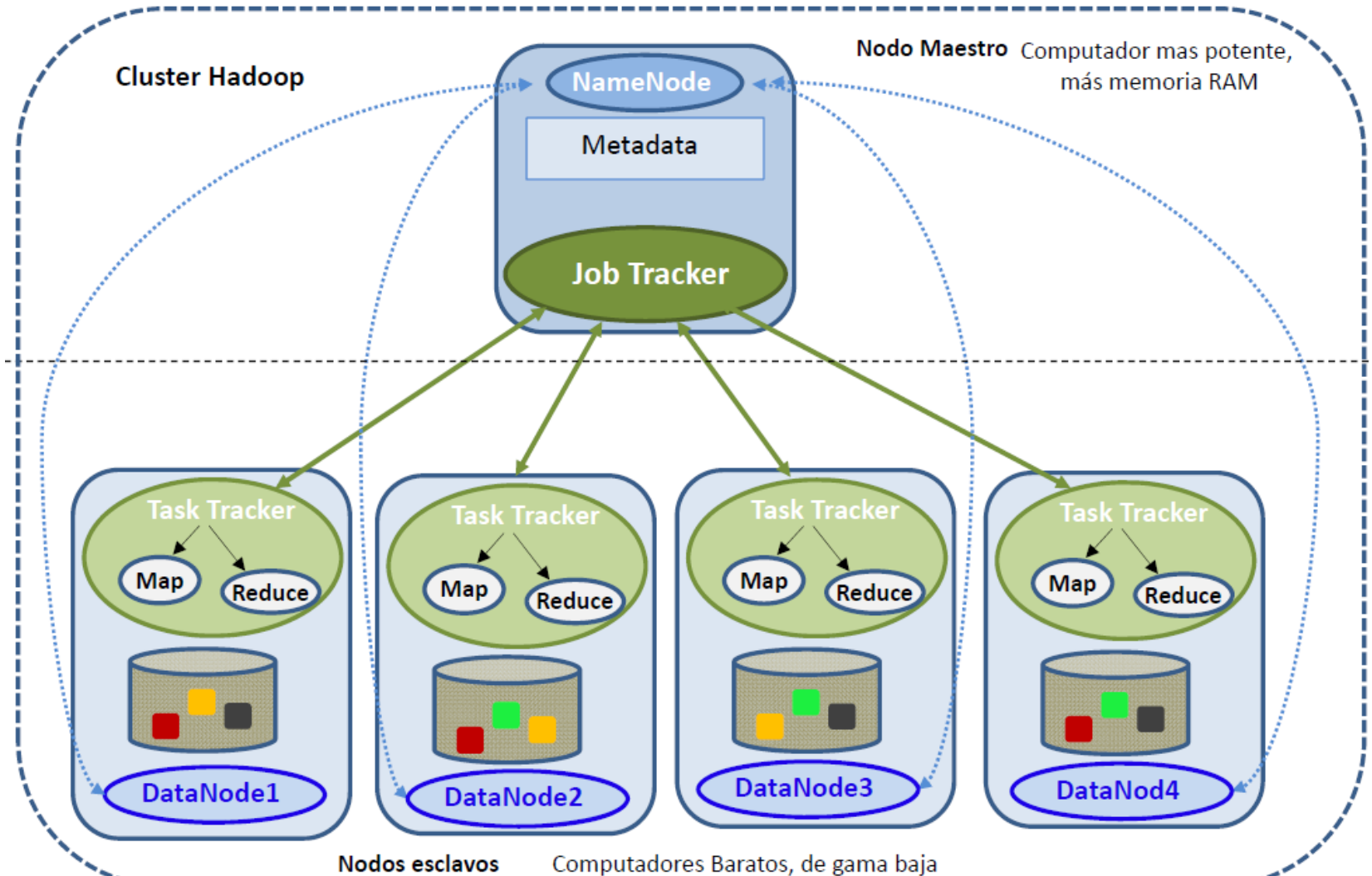


CTIC UNI

Centro de Tecnologías de Información y Comunicaciones
Universidad Nacional de Ingeniería

Hadoop es la implementación open source de
MapReduce

Implementación MapReduce



Implementación MapReduce



CTIC UNI

Centro de Tecnologías de Información y Comunicaciones
Universidad Nacional de Ingeniería

Tolerancia a fallos

- Si falla una tarea map o reduce
 - La tarea se reintenta en otro nodo
 - Sin problemas para los maps porque no tienen dependencias
 - Sin problemas para los reduce porque las salidas de los maps están en disco.
 - Necesario que las tareas no tengan efecto colaterales
 - Si la falla es repetida, el job se aborta.
- Si falla un nodo completo
 - Relanzan sus tareas en otros nodos
 - Reejecuta las tareas map completadas en ese nodo
 - Las tareas reduce completadas fueron guardadas en un sistema de archivos global.

Implementación MapReduce



CTIC UNI

Centro de Tecnologías de Información y Comunicaciones
Universidad Nacional de Ingeniería

Tolerancia a fallos

- Si falla el MasterNode
 - Se reintenta relanzar
 - Si continua, el trabajo se aborta y se notifica al usuario

Implementación MapReduce



Master

- Estado de cada tarea: *en-reposo* , *en-progreso*, *completada*
- Las tareas *en-reposo* se planifican a medida que van quedando workers libres
- Cuando una tarea map acaba, envía al Master la localización y tamaño de sus ficheros intermedios (uno por cada tarea reduce)
- El Master envía esta información a las tareas reduce, cuando acaban todos los map

Ejercicio



CTIC UNI

Centro de Tecnologías de Información y Comunicaciones
Universidad Nacional de Ingeniería

En equipo contesta las siguientes preguntas:

1. ¿Cómo están compuesto los recurso humanos, organizacionales y tecnológicos dónde trabajas o estudias?
2. ¿Con qué recursos mejorarías tu organización?
3. En qué etapa está la empresa respecto a la evolución de una Data-driven Company



CTIC UNI

Centro de Tecnologías de Información y Comunicaciones
Universidad Nacional de Ingeniería

Preguntas

Bibliografía



CTIC UNI

Centro de Tecnologías de Información y Comunicaciones
Universidad Nacional de Ingeniería

<http://udacity.com/>

<https://udemy.com/>

<https://cognitiveclass.ai/>



CTIC UNI

Centro de Tecnologías de Información y Comunicaciones
Universidad Nacional de Ingeniería

Tecnologías para el Big Data