



UAB

Universitat Autònoma
de Barcelona

Tomás Margalef

Modelo MapReduce

Introducción al Modelo MapReduce



MapReduce



YARN Yet Another Resource Negotiator



HDFS Hadoop Distributed File System



Hardware Básico



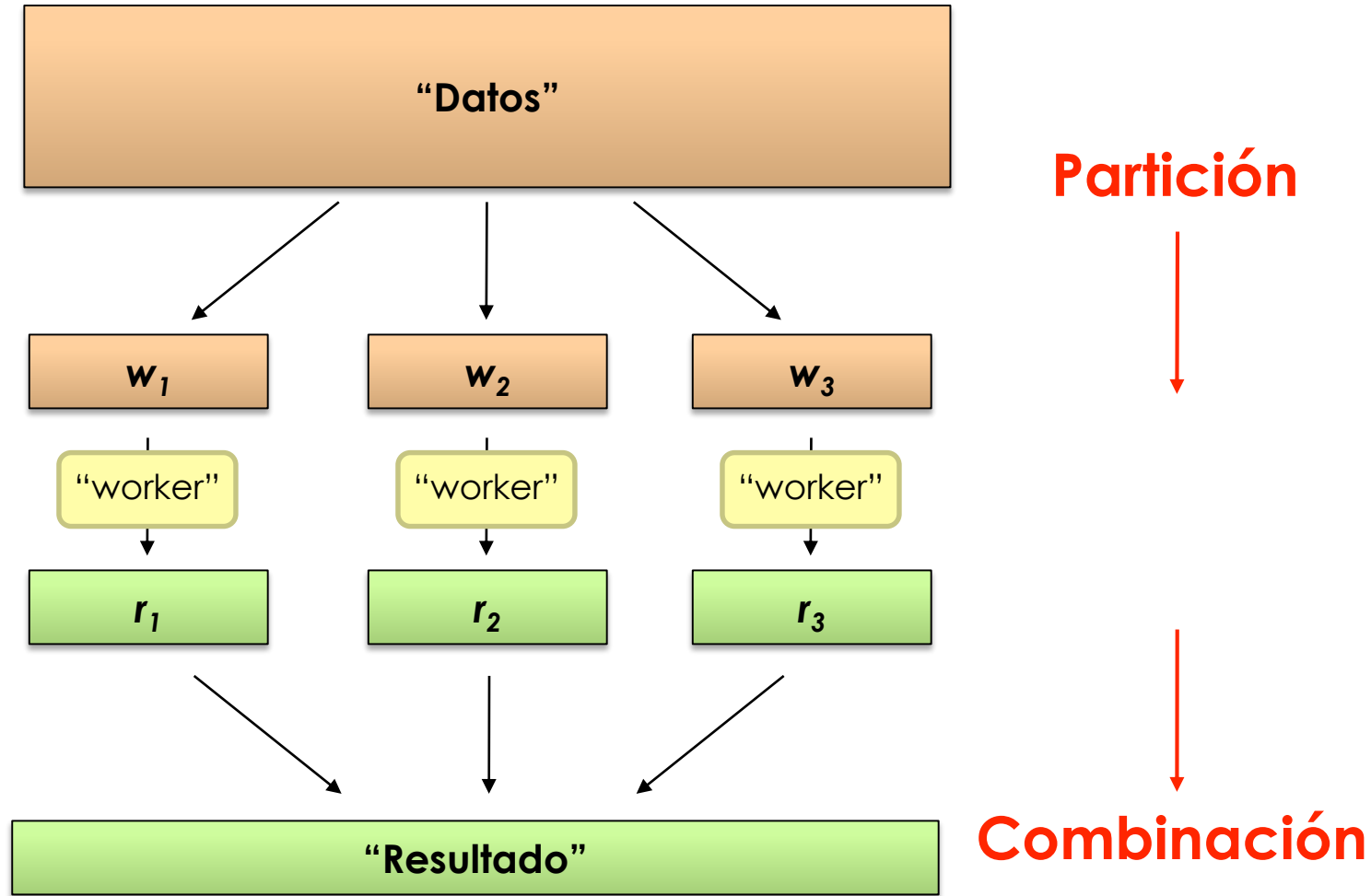
Introducción al Modelo **MapReduce**

Problema típico en Big Data

- Operar sobre un gran número de registros.
- Extraer información relevante de cada uno de ellos.
- Combinar y ordenar resultados intermedios.
- Agregar resultados intermedios.
- Generar los resultados de salida finales.



Introducción al Modelo MapReduce



Características de MapReduce (I)

- MapReduce permite procesar grandes cantidades de datos en forma paralela.
- MapReduce puede explotar las características de los sistemas de cómputo paralelos/distribuidos.
- MapReduce es un “*framework*” altamente escalable.
- MapReduce divide el procesamiento en dos fases fundamentales: La fase de mapeo “*Map*” y la fase de reducción “*Reduce*”.



Características de MapReduce (II)

- Los procesos que ejecutan la fase de mapeo se denominan “*Mappers*”. Los *Mappers* generalmente se ejecutan en los nodos en los que se encuentran los datos que van a procesar.
- El número de *Mappers* viene fijado por el framework, no por el desarrollador.
- Los *Mappers* realizan operaciones sobre los datos y devuelven pares *clave-valor* a la siguiente fase, la ordenación (“*sort and shuffle*”).

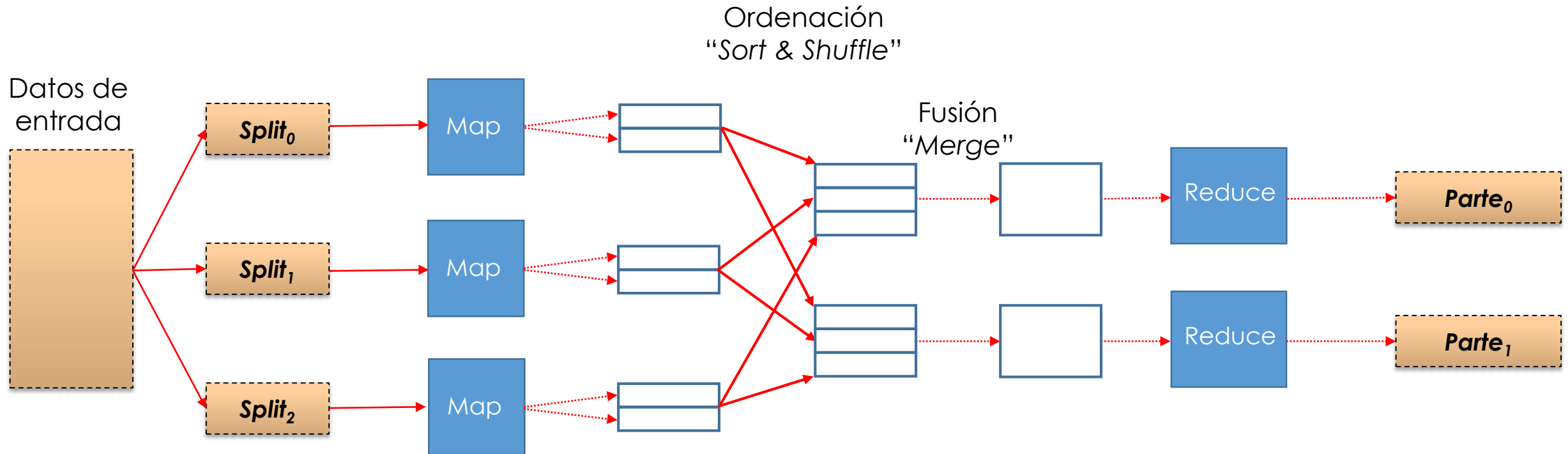


Características de MapReduce (III)

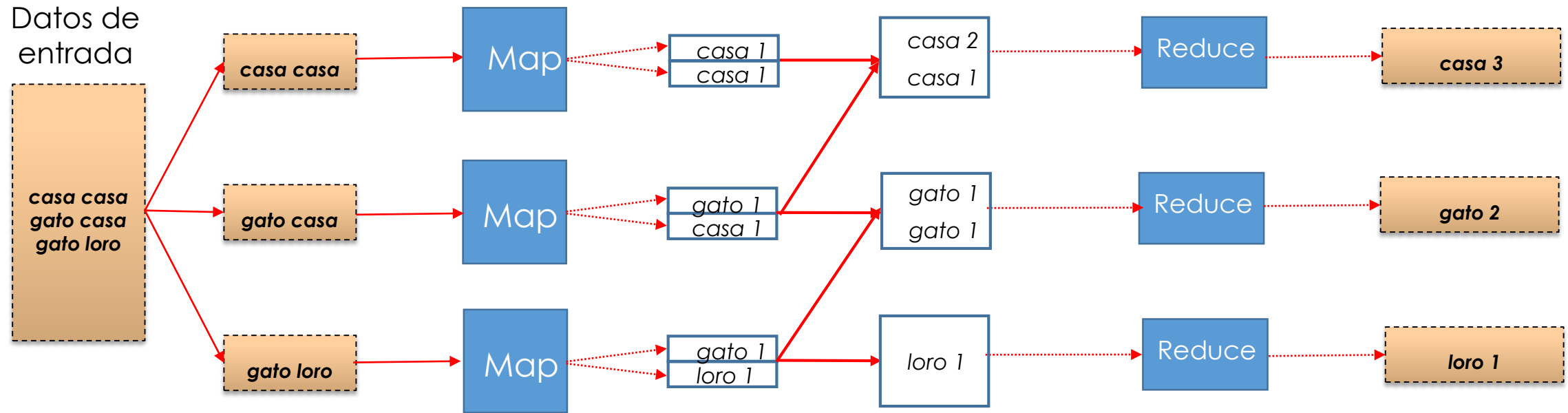
- En la fase de ordenación los datos se ordenan y particionan de acuerdo con las claves obtenidas por los *Mappers*.
- Los datos particionados y ordenados se envían a los procesos reductores “*Reducers*”.
- Los *Reducers* ejecutan la fase de reducción en la que se pueden realizar distintas operaciones sobre los datos.



Modelo MapReduce



Contador de palabras



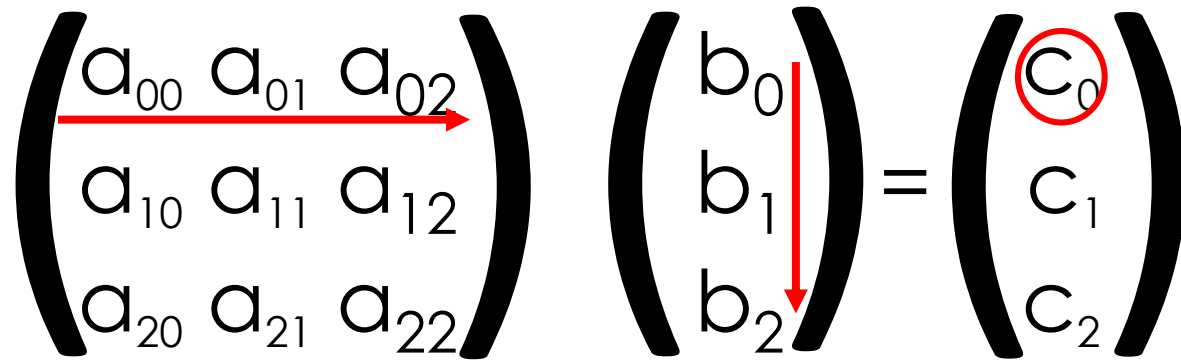
Contador de palabras

```
Map(String docid, String text):  
    for each word w in text:  
        Emit(w, 1);
```

```
Reduce(String term, Iterator <Int> values):  
    int sum = 0;  
    for each v in values:  
        sum += v;  
    Emit(term, sum);
```



Multiplicación Matriz-Vector

$$\begin{pmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} c_0 \\ c_1 \\ c_2 \end{pmatrix}$$


$$c_i = \sum a_{ij} * b_j$$



Multiplicación Matriz-Vector


Map(Matriz a, Vector b):
for each position i in b:
Emit(i, $a_{ij} * b_j$);

Reduce(String term, Iterator <Int> values):
int sum = 0;
for each v in values:
sum += v;
Emit(term, sum);



Modelo MapReduce



- Es un paradigma de programación sencillo.
- Permite explotar el paralelismo para el análisis y procesamiento de los datos.
- Permite realizar distintos tipos de operaciones sobre los datos a procesar.
- Es un modelo adecuado de procesamiento que ha sido adoptado por los entornos Big Data como 





Modelo MapReduce



Universitat Autònoma
de Barcelona