

UNIDAD 2: MODELO DE MAPEO Y REDUCCIÓN

MODELO COSTO - COMUNICACIÓN

Blanca Vázquez

Febrero 2020

INTRODUCCIÓN

Una de las características principales del modelo de programación MapReduce es la paralelización de las tareas

Mapeo
Lee una entrada
y produce un
conjunto de pares
llave - valor

**Agrupar por
llaves:**
colecciona todos
los pares con la
misma llave

Reducir:
colecciona todos
los valores que
pertenecen a la
llave

Hoy empecé la dieta verde:
verde lejos la pizza,
verde lejos los tamales,
verde lejos las tortas,
verde lejos el pan.

(hoy, 1)
(empece, 1)
(la, 1)
(dieta, 1)
(verde, 1)
(verde, 1)
(lejos, 1)
(la, 1)
(pizza, 1)
(verde, 1)
(lejos, 1)
(los, 1)
(tamales, 1)
(verde, 1)
(lejos, 1)
(las, 1)
(tortas, 1)
(verde, 1)
(lejos, 1)
(el, 1)
(pan, 1)

(llave, valor)

(hoy, 1)
(empece, 1)
(la, 1)
(la, 1)
(dieta, 1)
(verde, 1)
(verde, 1)
(verde, 1)
(verde, 1)
(verde, 1)
(lejos, 1)
(lejos, 1)
(lejos, 1)
(lejos, 1)
(pizza, 1)
(lejos, 1)
(tamales, 1)
(las, 1)
(tortas, 1)
(el, 1)
(pan, 1)

(llave, valor)

(hoy, 1)
(empece, 1)
(la, 2)
(dieta, 1)
(verde, 5)
(lejos, 4)
(pizza, 1)
(los, 1)
(tamales, 1)
(las, 1)
(tortas, 1)
(el, 1)
(pan, 1)

(llave, valor)

Únicamente lecturas secuenciales

El costo de comunicación de un algoritmo es **la suma del costo de la comunicación de todas las tareas implementadas en el algoritmo.**

- **Robustez:** es la capacidad de un programa para detectar condiciones excepcionales (tratarlas o cerrarlas)
- **Exactitud:** el programa hace *'lo que se supone que debe hacer'*
- **Eficiencia:** se refiere a todos los recursos que usa un programa (tiempo, espacio y energía)

De manera general, el rendimiento de un algoritmo puede ser descrito en los siguientes costos:

- **Tiempo:** ¿cuánto tiempo lleva en ejecutar una tarea? *entre menos tiempo es mejor*
- **Espacio:** ¿cuánta memoria usa? *entre menos espacio es mejor*
- **Energía:** ¿cuánta energía usa? *entre menos energía es mejor*

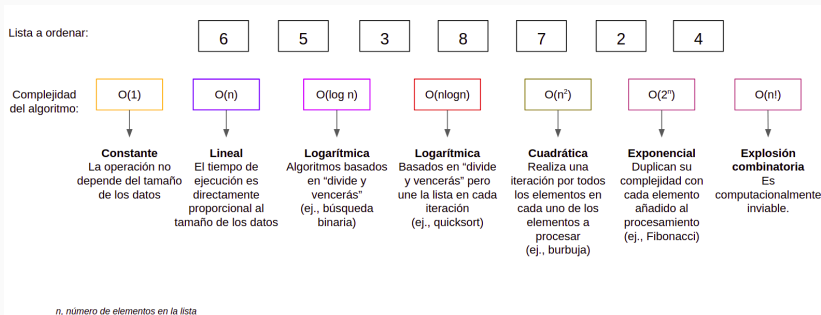
Para un algoritmo dado, el objetivo es encontrar las siguientes funciones:

- $T(n)$: el costo del tiempo para resolver el problema
- $S(n)$: el costo del espacio para resolver el problema
- $E(n)$: el costo de la energía para resolver el problema

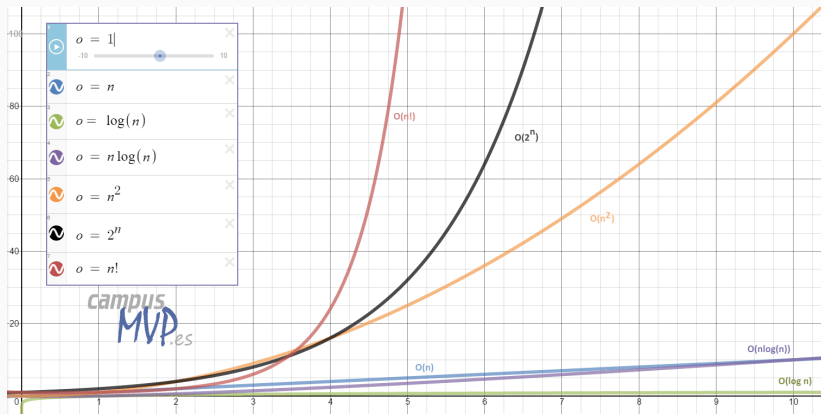
La idea básica para calcular los costos de un algoritmo es:

- Implementar el algoritmo
- Medir tiempo, espacio y energía usados en la ejecución del algoritmo
- Graficar las mediciones $T(n)$, $S(n)$, $E(n)$

COMPLEJIDAD DE LOS ALGORITMOS



COMPLEJIDAD DE LOS ALGORITMOS



El costo de un algoritmo depende de varias características:

- La semántica del modelo de programación
- La topología de la red
- El tipo de datos
- Protocolos de comunicación

COSTO DE LOS ALGORITMOS EN MAPREDUCE

En MapReduce el costo de un algoritmo se calcula de la siguiente manera:

- **Costo de comunicación** = es la suma total de entradas y salidas de todos los procesos
- **Costo de comunicación transcurrido** = es el tamaño máximo de las entradas y salidas para cualquier camino
- **Costo de computación transcurrido** = cuenta solo el tiempo de los procesos en ejecución

Recordemos que la complejidad de los algoritmos se puede resolver, en este esquema, añadiendo más nodos

Para un algoritmo usando el modelo MapReduce:

- **Costo de comunicación** = tamaño del archivo de entrada + (suma de todos los archivos pasados del proceso de mapeo hacia el proceso de reducción) + la suma de los tamaños de los archivos de salida en los procesos de reducción.

Para un algoritmo usando el modelo MapReduce:

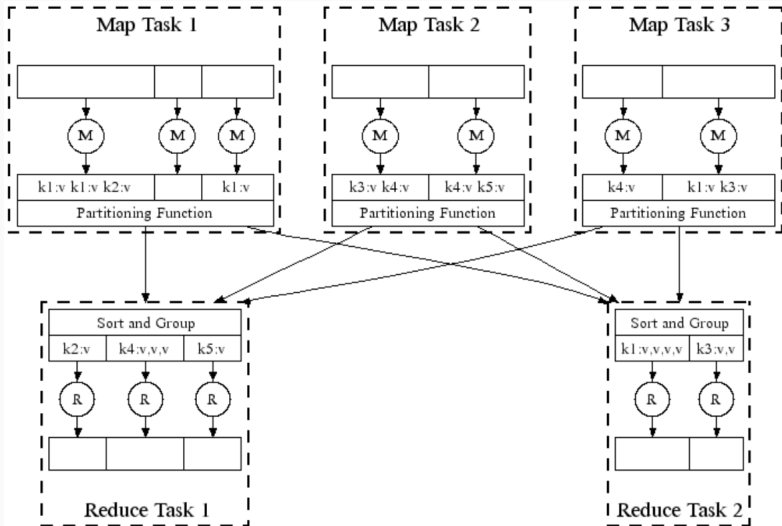
- **Costo de comunicación transcurrido** = es la suma de la salida y de la entrada más grande para el proceso de mapeo + la suma de la salida y de la entrada más grande para el proceso de reducción

Para un algoritmo usando el modelo MapReduce:

- **Costo de computación transcurrido** = suma únicamente el tiempo de ejecución de los procesos

- El costo total nos indica el pago de *renta* por hacer uso de una infraestructura en clúster
- El costo de comunicación transcurrido es el tiempo reloj usado en paralelismo.

MODELO MAPREDUCE EN PARALELO



EJEMPLO: UNA CONSULTA TRIANGULAR

- Input:** three tables
 $R(X, Y), S(Y, Z), T(Z, X)$

$$\text{size}(R) + \text{size}(S) + \text{size}(T) = M$$

- Output:** compute
 $Q(x,y,z) = R(x,y) \bowtie S(y,z) \bowtie T(z,x)$

		T	
		Z	X
		Fred	Alice
S		Y	Z
		Fred	Alice
			Jim
			Jim
R		X	Y
		Fred	Alice
		Jack	Jim
		Fred	Jim
		Carol	Alice
		...	

Imagen tomada de Dan Suciu, University of Washington

M = tamaño de datos de entrada en bits

- Paso 1: calcular un join (almacenarlo en temp)

$$temp(X, Y, Z) = R(X, Y) \bowtie S(Y, Z)$$

- Paso 2: calcular un segundo join

$$Q(X, Y, Z) = temp(X, Y, Z) \bowtie T(Z, X)$$

Cada *join* fue implementado por MapReduce

- 'temp' puede generar largas relaciones intermedias
- El tiempo de comunicación y computación transcurrido puede verse afectadas (dependerán del tamaño de temp)

EJEMPLO: CONSULTA TRIANGULAR EN UN SOLO JOIN

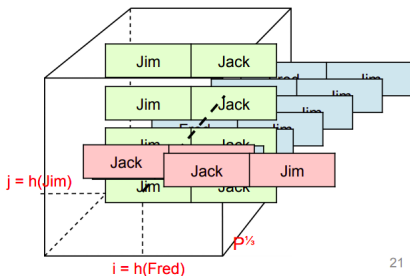
				T	
				z	x
				Fred	Alice
				S	
				y	z
				Fred	Alice
R	x	y		Jim	Jim
	Fred	Alice		Jim	Alice
	Jack	Jim			
	Fred	Jim		Jack	
	Carol	Alice			
	...				

Round 1:

Send $R(x,y)$ to all servers ($h(x), h(y), *$)
 Send $S(y,z)$ to all servers ($*, h(y), h(z)$)
 Send $T(z,x)$ to all servers ($h(x), *, h(z)$)

Output:

compute locally $R(x,y) \bowtie S(y,z) \bowtie T(z,x)$



21

Imagen tomada de Dan Suciu, University of Washington

- CSE 12 Algorithm Time Cost Measurement:

<http://cseweb.ucsd.edu/~kube/cls/12.s13/Lectures/lec06/lec06.pdf>

- Rendimiento de algoritmos y notación Big-O

<https://www.campusmvp.es/recursos/post/Rendimiento-de-algoritmos-y-notacion-Big-0.aspx>

- Communication Cost in Big Data Processing

http://mmds-data.org/presentations/2014/suciu_mmds14.pdf