

# ANÁLISIS DE DATOS MASIVOS

## MODELO COSTO-COMUNICACIÓN Y TEORÍA DE LA COMPLEJIDAD

---

Blanca Vázquez

20 de marzo de 2025

- 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

# COSTOS EN EL MODELO DE MAPEO Y REDUCCIÓN (1)

## 1. Costo de cómputo

- Trabajos de mapeo
- Trabajos de reducción

## 2. Costo de comunicación

- Transmisión de los pares llave-valor de los nodos de mapeo a los de reducción.
- En las tareas de mapeo usualmente no se incurre en costos de comunicación (se busca llevarlas a cabo en el mismo nodo donde se encuentran los datos, aunque se puede incurrir en costos de lectura de los datos de disco).

## COSTOS EN EL MODELO DE MAPEO Y REDUCCIÓN (2)

- En un escenario común
  - El costo de comunicación domina al de cómputo
  - El costo de los trabajos de mapeo es una pequeña fracción del costo de comunicación
  - El costo que incurre el sistema por el ordenamiento es proporcional al costo de comunicación
- En los servicios de la nube (por ej. EC2) se paga tanto por el cómputo como por la comunicación, por lo que es importante tener un balance de ambos en el diseño de los algoritmos

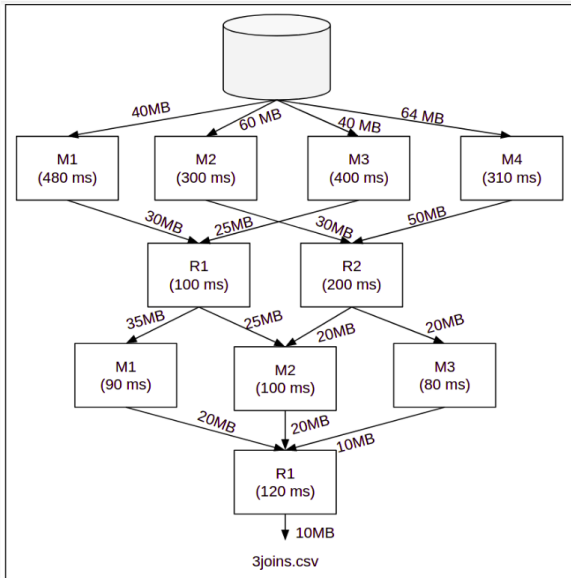
En MapReduce el costo de un algoritmo se calcula de la siguiente manera:

- Costo de comunicación
- Costo de comunicación transcurrido
- Costo de computación transcurrido

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

- Costo de comunicación: es la suma de IO de todos los procesos.
- Costo de comunicación transcurrido: es el valor máximo de las I/O de las tareas de mapeo y de reducción.
- Costo de computación transcurrido: suma todos\* los tiempos de ejecución

# EJERCICIO: MODELO COSTO COMUNICACIÓN





# MODELO DE COSTO DE COMUNICACIÓN

- El costo de comunicación de un algoritmo es la suma de los costos de todas las tareas que requiere
- La meta es medir la eficiencia de un algoritmo basado en su costo de comunicación
  - El cómputo que realiza cada tarea es usualmente muy simple y con complejidad lineal en el tamaño de la entrada.
  - Transmitir una entrada sobre la red puede ser muy lento y requerir tiempo par poder recibirla de otros nodos debido al tráfico en la red.

- 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

- Al diseñar un algoritmo para un clúster es importante también tomar en cuenta el tiempo que toma un algoritmo en ejecutarse en paralelo.

- En MapReduce, el paralelismo se genera debido a que cada tarea de mapeo y reducción se ejecutan al mismo tiempo en diferentes nodos.
- El programador define el número de nodos para cada tarea.
- El modelo se encarga de planificar los pasos de mapeo, agrupación, reducción, rutas de los datos y tolerancia a fallos.

- Un *round* consiste de las tareas de **mapear - agrupar - reducir**.
- La salida de una tarea de reducción puede ser la entrada a otro round.
- En aplicaciones reales, MapReduce es una secuencia de tareas que se ejecutan a través de *rounds*.
- Idealmente se espera que el número de *rounds* sea una constante.

## 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     |
| S | Y     | Fred  | Alice |
|   |       |       | Jim   |
|   | Z     | Alice | Jim   |
|   |       |       | Alice |
| R | X     | Y     | Jim   |
|   |       |       | Alice |
|   | Fred  | Alice | Jim   |
|   | Jack  | Jim   | Alice |
|   | Fred  | Jim   |       |
|   | Carol | Alice |       |
|   | ...   |       |       |
|   |       |       |       |
|   |       |       |       |
|   |       |       |       |

Imagen tomada de Dan Suciu, University of Washington

M = tamaño de datos de entrada en bits

## HACIENDO LA UNIÓN EN DOS PASOS

- Paso 1: calcular primera unión (almacenarlo en *temp*)  
 $temp(X, Y, Z) = R(X, Y) \bowtie S(Y, Z)$
- Paso 2: calcular una segunda unión  
 $Q(X, Y, Z) = temp(X, Y, Z) \bowtie T(Z, X)$

Nota:

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

Para medir la eficiencia del algoritmo MapReduce sobre el transcurso de su ejecución debemos obtener:

- $R$ : número de rounds que el algoritmo usará
- $C_r$ : la complejidad de comunicación del round  $r$ .
- $t_r$ : tiempo de ejecución interna



## COMPLEJIDAD DE COMUNICACIÓN $C_r$ DEL ROUND $r$

Denotamos  $n_{r,i}$  como el tamaño de las I/O de las tareas de mapeo y reducción  $i$  en el round  $r$ .

$$C_r = \sum_i n_{r,i} \text{ complejidad de comunicación del round } r$$
$$C = \sum_{r=0}^{R-1} C_r \text{ complejidad de comunicación para todo el algoritmo}$$

## CÁLCULO DEL TIEMPO DE EJECUCIÓN INTERNO $t_r$ PARA CADA ROUND $r$

Denotamos como  $t_r$  como el tiempo máximo de ejecución interno para la tarea de mapeo o reducción en el round  $r$ .

$$t = \sum_{r=0}^{R-1} t_r \text{ tiempo total de ejecución interno}$$

Para calcular la eficiencia total de MapReduce debemos considerar:

- **Latencia de la red (L):** se refiere al número de pasos que la tarea de mapeo o reducción tienen que esperar para recibir su 1era entrada en una ronda.
- **Ancho de banda (B):** es el número de elementos que puede ser entregado por la red en una unidad de tiempo.

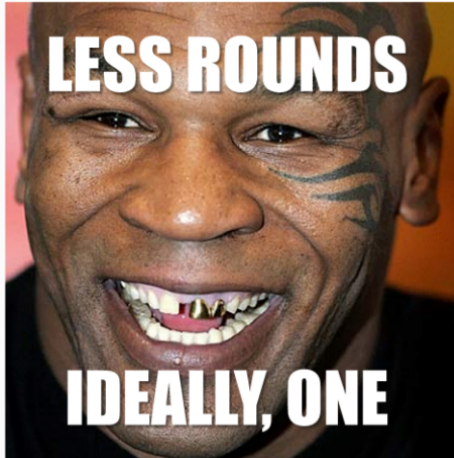
El tiempo total de ejecución para la implementación de un algoritmo se define como:

$$T = \Omega(\sum_{r=0}^{R-1}(t_r + L + C_r/B)) = \Omega(t + RL + C/B)$$

$\Omega$  representa el conjunto de todas las funciones de complejidad para un algoritmo (cota inferior).

## EL NÚMERO DE ROUNDS ES CLAVE

El número de *rounds* está proporcionalmente relacionado con el tiempo total de comunicación y la complejidad del algoritmo.



- Si nos enfocamos a tener un solo round, tendríamos un algoritmo ineficiente.
- Comunicación limitada: existe una estricta modularidad que prohíbe que los nodos de reducción se comuniquen entre sí.
- Porción limitada de datos: para cada nodo de mapeo y de reducción.
- Dado lo anterior, frecuentemente hay más de round.

- Limitar la tasa de replicación
- Cálculo de la complejidad previo al algoritmo
- Uso de grafos