

Método del Trapecio

Motivación: Un nuevo método.

Motivación

- Ecuaciones diferenciales ordinarias.
- ¿Existe solución y es única?
- Métodos de discretización.
- Método de Euler.

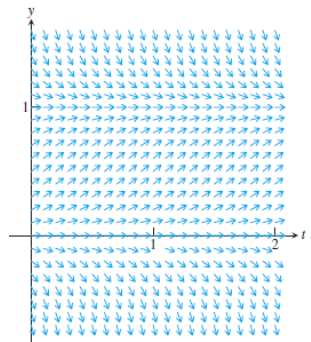


Figura: Representación del campo vectorial asociado a la ecuación logística $y'(t) = cy(t)(1 - y(t))$.

Motivación: Método de Euler

$$\begin{cases} w_0 = y_0 \\ h_i = t_{i+1} - t_i \\ w_{i+1} = w_i + h_i f(t_i, w_i) \end{cases} \quad (1)$$

Método de Euler

- Mejores resultados para puntos equidistantes.
- Es estable, consistente y convergente.
- El error global de aproximación es $O(h)$.
- Puede parecer válido en cualquier aplicación.

Motivación: Ejemplo

Considérese el siguiente problema de valores iniciales:

$$\begin{cases} y'(t) = -4t^3 y^2 \\ y(-10) = 1/10001 \\ t \in [-10, 0] \end{cases}$$

- La solución exacta es $y(t) = \frac{1}{1+t^4}$.
- Queremos calcular la aproximación de y en 0 con $y(0) = 1$.
- Se va a aproximar hasta llegar a los 10000 puntos.

Motivación: Ejemplo

N	h	w_n
100	0.1	0.00390138
1000	0.01	0.03085162
5000	0.002	0.13282140
7500	0.0013	0.18614311
10000	0.001	0.23325153

Tabla: Ejemplo de un mal comportamiento del método de Euler.

Motivación: Ejemplo

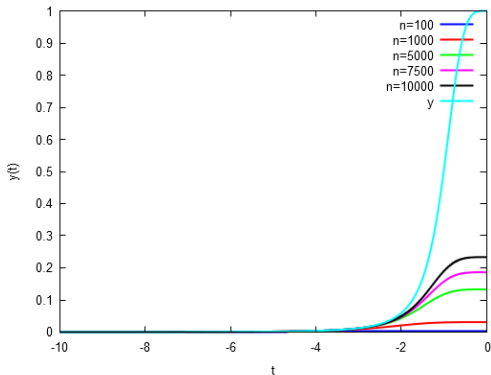


Figura: Aproximaciones obtenidas con diferentes valores de n .

Índice

- 1 Motivación
- 2 TPCx-HS
- 3 Método del trapecio
Introducción
- 4 Conclusión

¿Por qué usar TPCx-HS?

¿Qué es TPCx-HS?

TPC™

Transaction Processing Performance Council Express Hadoop System

Benchmarking Hadoop

Carga de trabajo de TPCx-HS

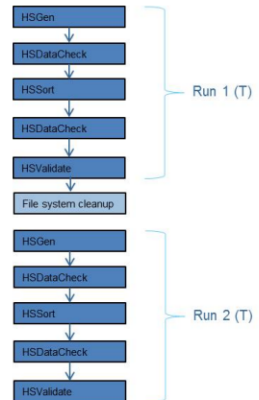
- HSGen: generación de datos con un factor de escala.
- HSDataCheck: comprobación de los datos.
- HSSort: Implementación en Hadoop de TeraSort.
- HSValidate: comprobación de la salida.



Funcionamiento de TPCx-HS

Dos ejecuciones de cinco fases cada una.

- Fase 1: Generación de los datos.
3-ways replication
- Fase 2: Verificación de la validez de los datos.
- Fase 3: Ordenación de los datos.
3-ways replication
- Fase 4: Verificación de la validez de los datos.
- Fase 5: Validación de la salida



Rendimiento

Medida del rendimiento.

$$HSph@SF = \frac{SF}{T/3600}$$

Medida del rendimiento-precio.

$$$/HSph@SF = \frac{P}{HSph@SF}$$

Parámetros:

- SF: factor de escala escogido.
- T: tiempo total de las dos ejecuciones.
- P: costo del sistema bajo estudio.

Introducción al método del trapecio

Proposición

Sea un PVI con $y'(t) = f(t, y(t))$ y $y(t_0) = y_0$. Son equivalentes:

- ① y es una solución del PVI.
- ② $y(t) = y_0 + \int_{t_0}^t f(s, y(s)) ds \quad \forall t \in [a, b]$

Nuestra solución verifica:

$$y(t_1) = y_0 + \int_{t_0}^{t_1} f(s, y(s)) ds$$

Introducción al método del trapecio

Nuestra solución verifica:

$$y(t_1) = y_0 + \int_{t_0}^{t_1} f(s, y(s)) ds$$

Idea: Método del trapecio para integración numérica

$$y(t_1) = y_0 + \frac{h}{2} [f(t_0, y_0) + f(t_1, y(t_1))] - \frac{h^3}{12} y^{(3)}(\xi) \quad (2)$$

$$y(t_1) \approx w_1 = w_0 + \frac{h}{2} [f(t_0, w_0) + f(t_1, y(t_1))] \quad (3)$$

¿Cómo funciona map reduce?

Algoritmo: Obtención del número de ocurrencias de cada una de las palabras de un texto.

key: Nombre del documento

value: Texto del documento

function MAP(String *key*, String *value*)

for each word *w* in *value* **do**

 EmitIntermediate(*w*, "1")

end for

end function

key: Palabra

values: Conjunto con las ocurrencias de la palabra

function REDUCE(String *key*, Iterator *values*)

result = 0

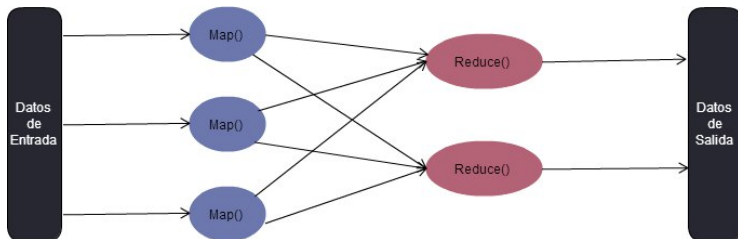
for each value *v* in *values* **do**

result += Int(*v*);

end for

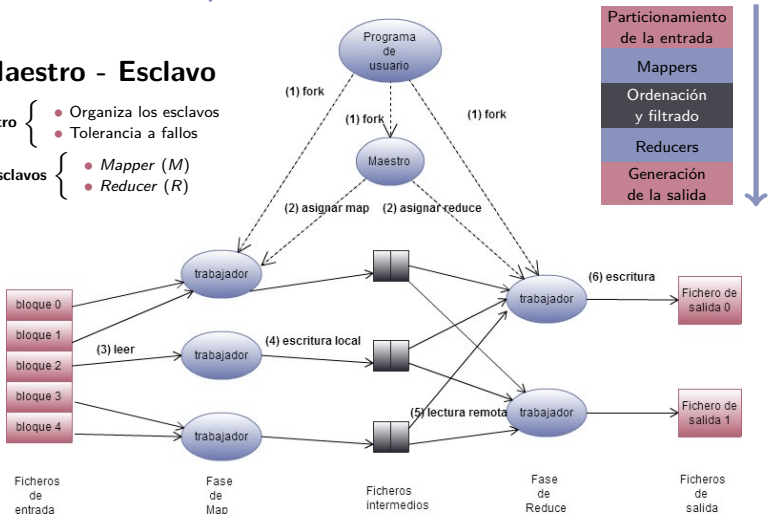
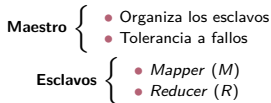
return *key*, String(*result*);

end function



¿Cómo funciona map reduce?

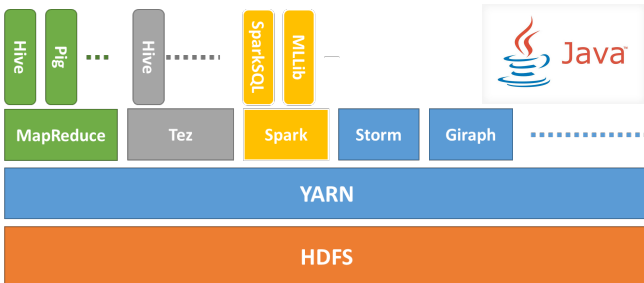
Maestro - Esclavo



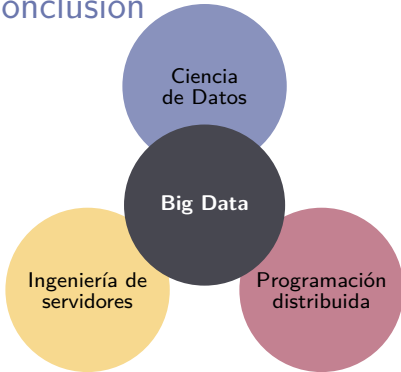


Open-source software for reliable, scalable, distributed computing

- HDFS: Sistema de archivos distribuido basado en Google File System (GFS).
- YARN: Gestión de tareas, recursos y nodos.
- SPARK: Map Reduce + procesamiento iterativo y en memoria.



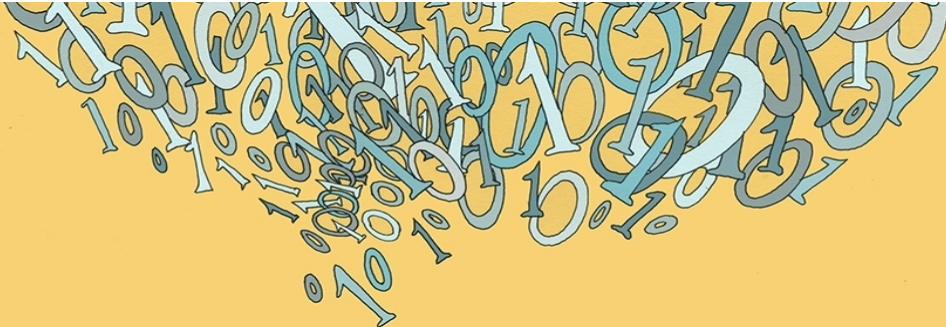
Conclusión



- Nuevas tecnologías: Spark, Flink...
- Desarrollo y diseño de algoritmos
- Benchmarks para las nuevas tecnologías

“Vivimos en la era de la información. El progreso y la innovación no se ve obstaculizado por la capacidad de recopilar datos sino por la capacidad de gestionar, analizar, sintetizar y descubrir el conocimiento subyacente en dichos datos. Este es el reto de las tecnologías de Big Data.”

Francisco Herrera Triguero, Prof. Universidad de Granada



Gracias por su atención.

Ilustración de Lola Moral y Sergio García