

Modelos Paralelos para la Resolución de Problemas de Ingeniería Agrícola

Defensa Tesis de Doctorado en Informática

Murilo Boratto



Pedro Alonso¹, Domingo Giménez²

¹Departamento de Sistemas Informáticos y Computación (DSIC)
Universidad Politécnica de València (UPV)

²Departamento de Informática y Sistemas (DIS)
Universidad de Murcia (UM)

València, Enero 2015

Contenido

Introducción y Contextualización de la investigación

Planteamiento, objetivos, ámbito del trabajo, problemas a tratar, estado del arte y herramientas computacionales

Problemas Aplicados a Ingeniería Agrícola

Representación del Relieve

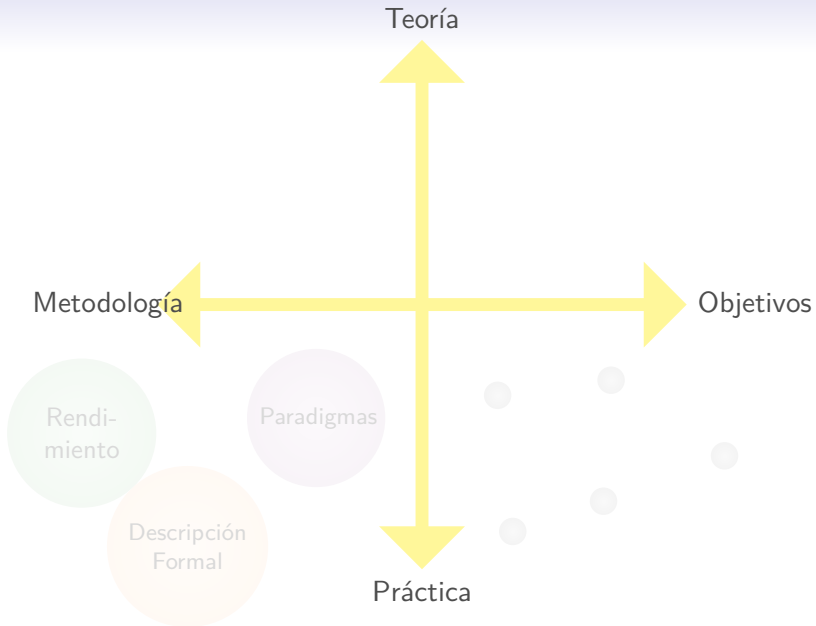
Polinomios Matriciales

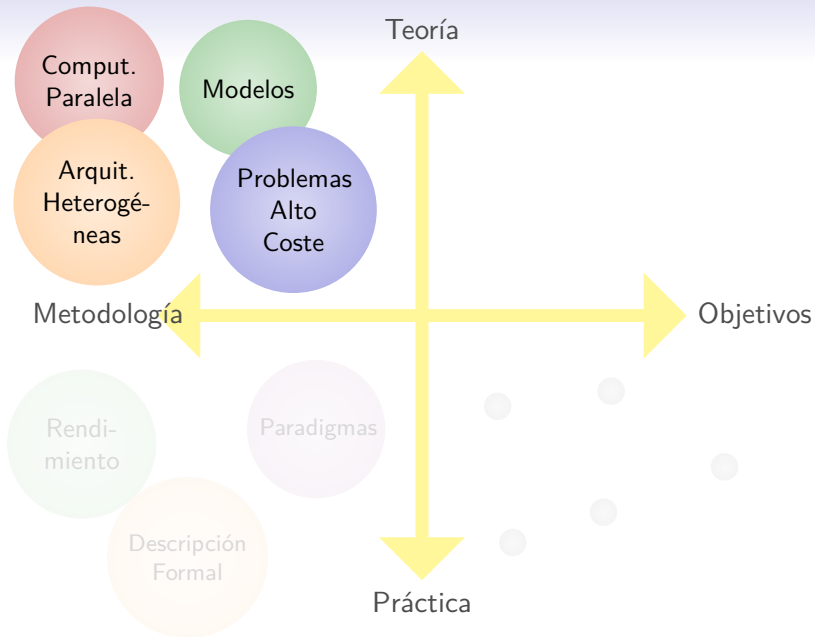
Variables Meteorológicas del Clima

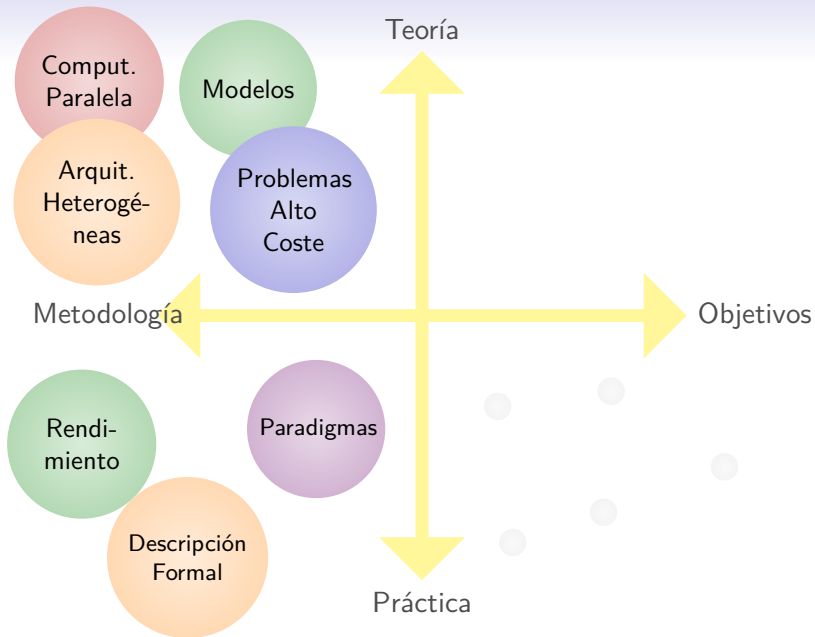
Modelado Hídrico de Ríos

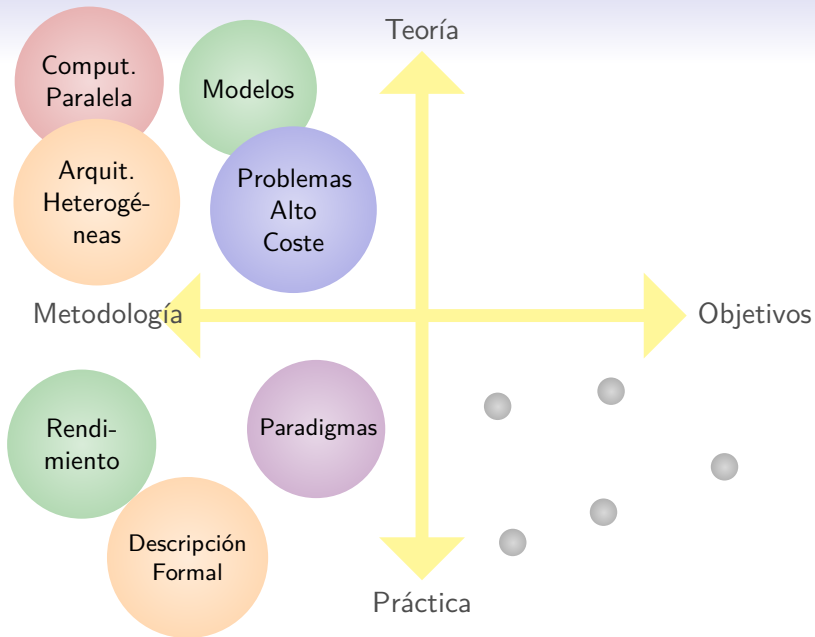
Autooptimización en Aplicaciones Paralelas

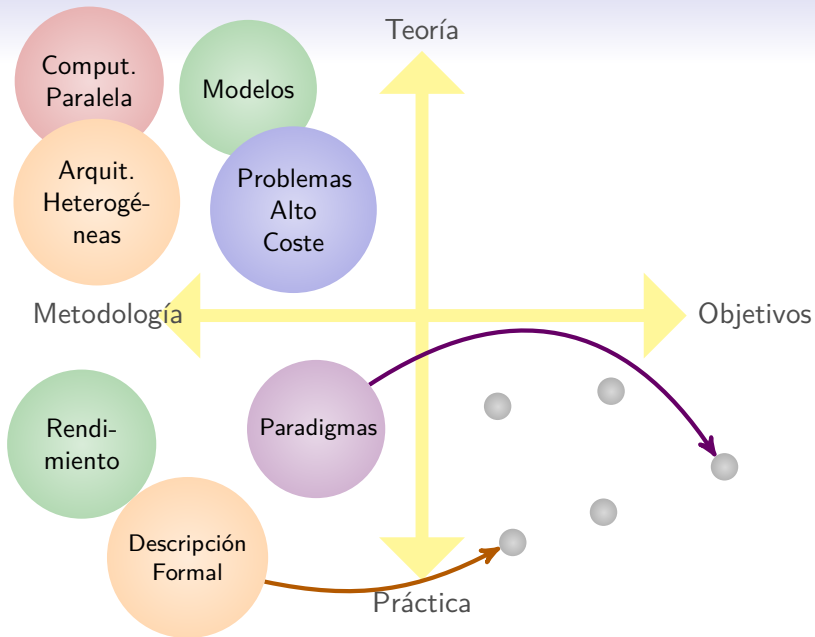
Conclusiones Generales y Aportaciones

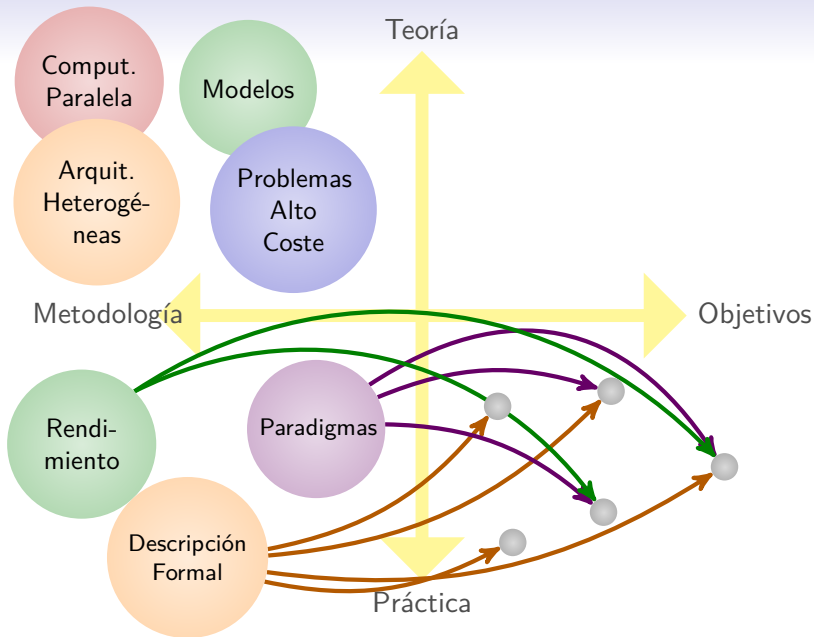


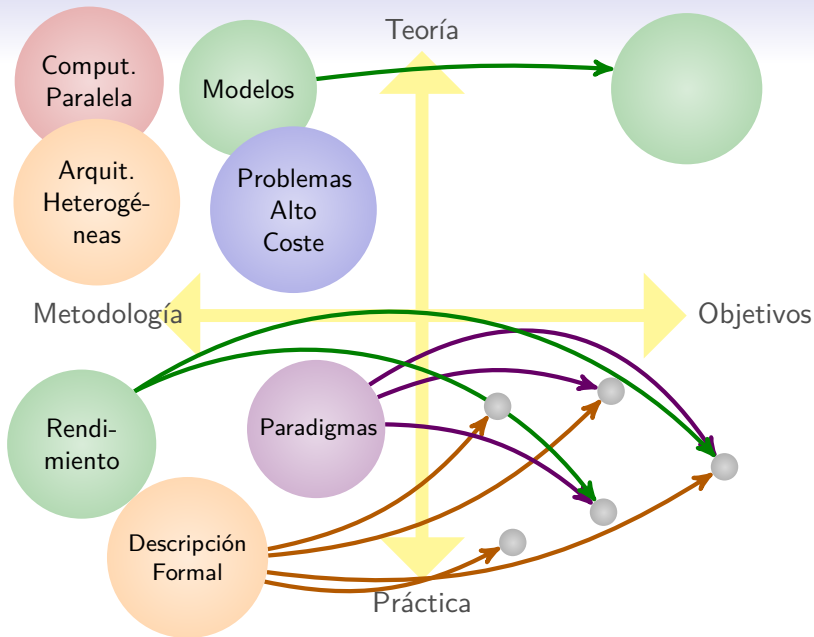


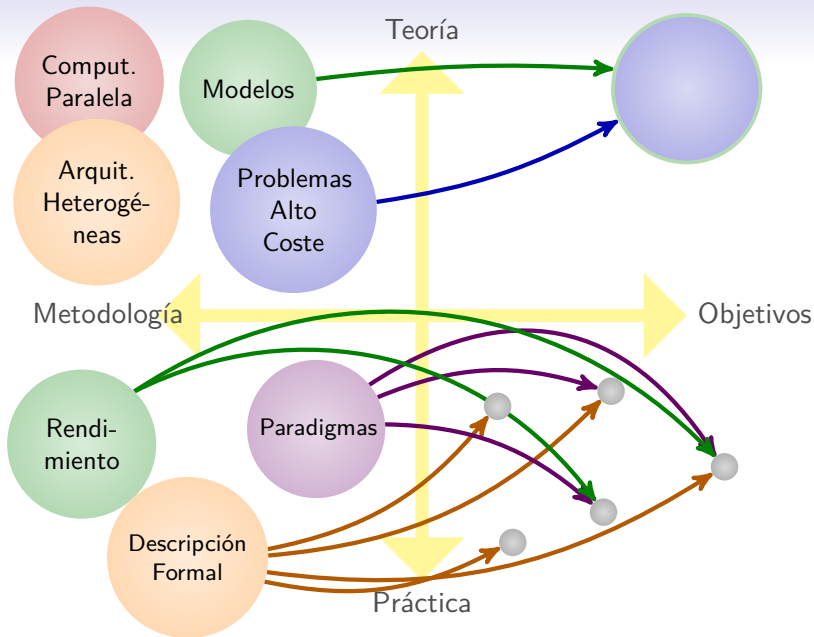


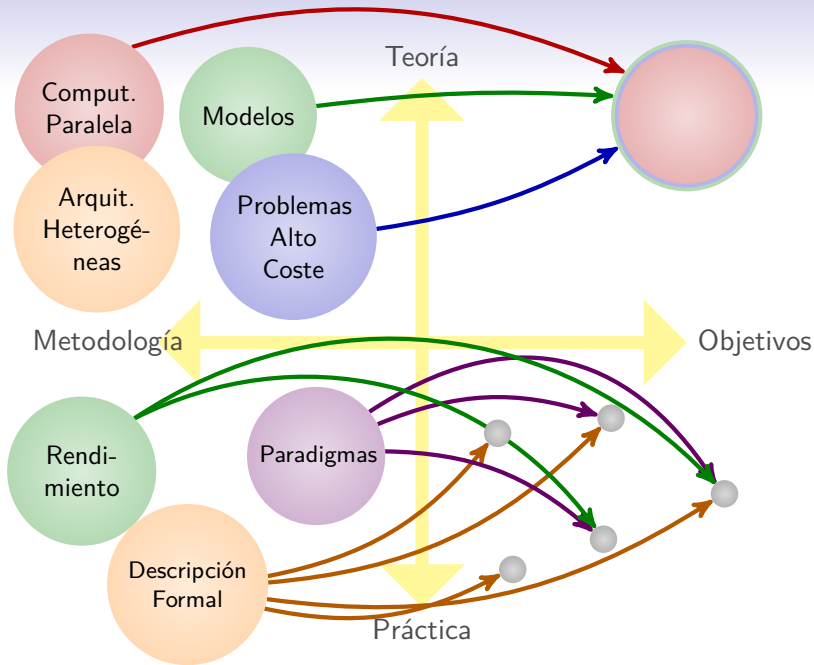


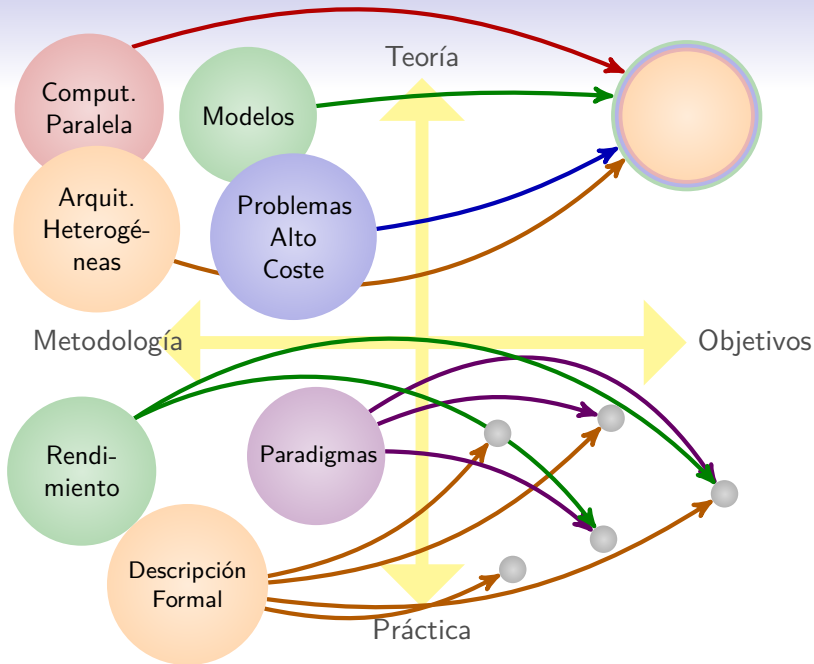


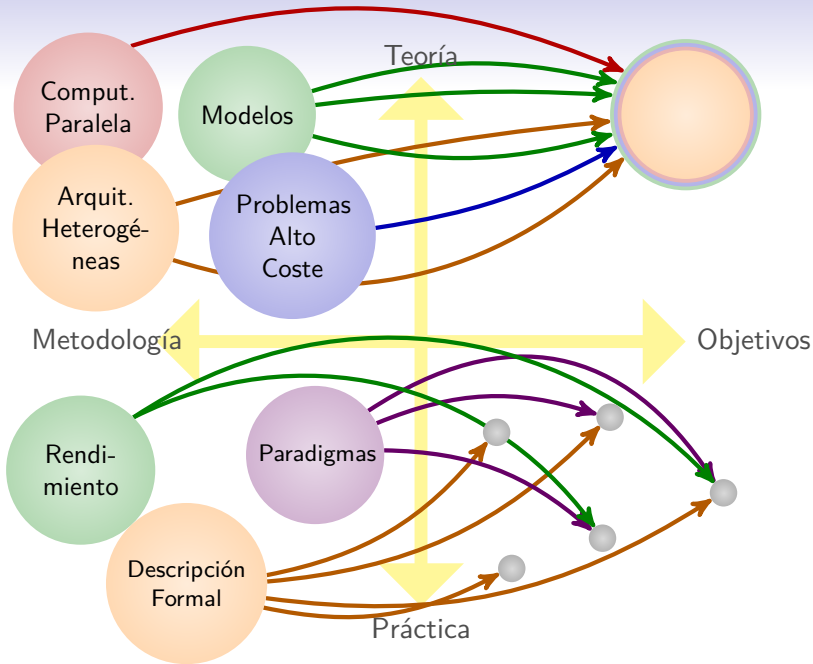


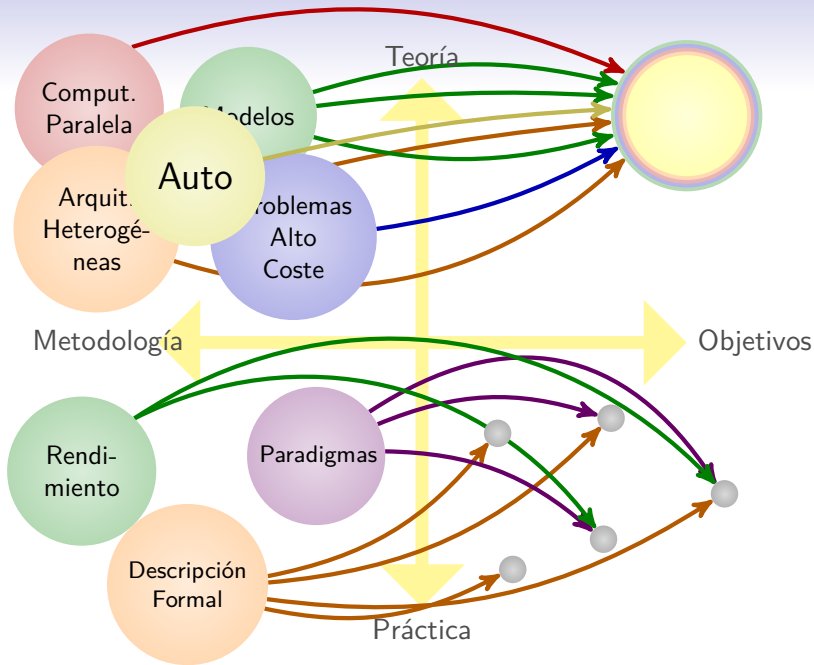


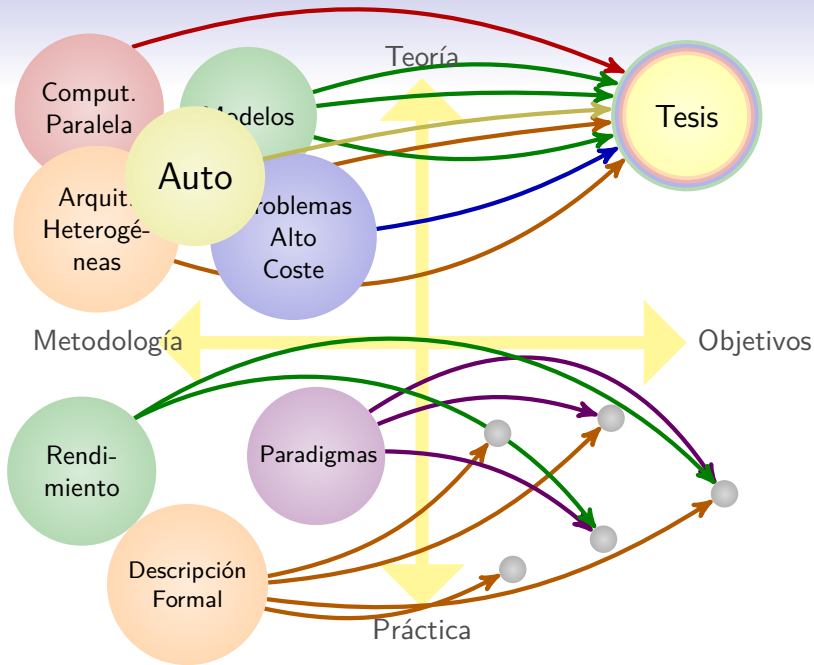












Objetivos Generales

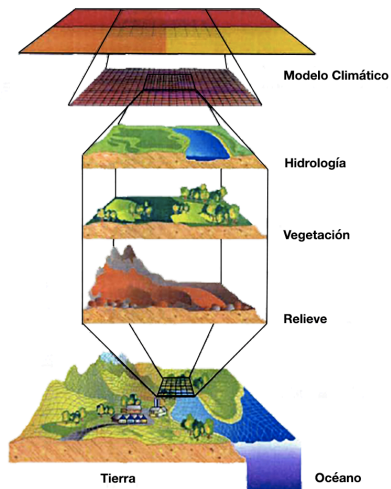
Los **objetivos generales** de nuestro proyecto de investigación se pueden resumir en los siguientes puntos:

- 1 Hacer funcionar la aplicación eficientemente en entornos **heterogéneos**.
- 2 Combinar los procesadores **multicore** y **multi-GPU**.
- 3 Aportar técnicas que permitan el desarrollo de **rutinas paralelas** que puedan adaptarse **automáticamente** a las características del sistema paralelo donde se ejecutan.

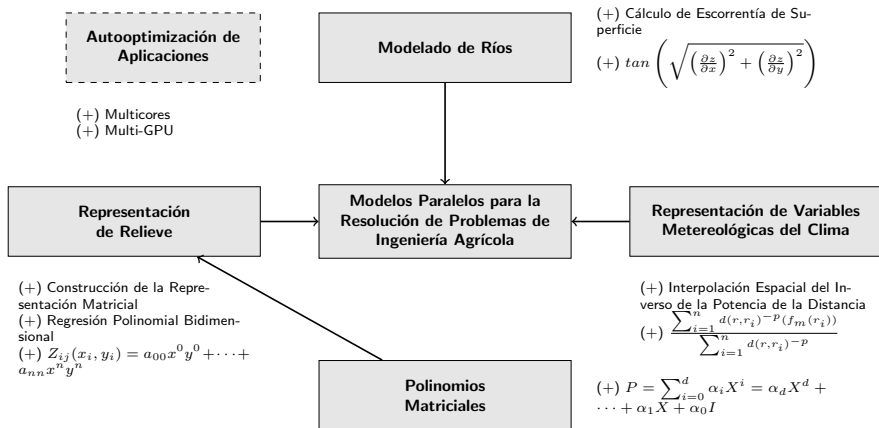
Los problemas agrícolas del Valle Rio São Francisco

Ámbito del trabajo

- **Centro de Computación Agrícola y de la Tierra**
- **Valle del Rio São Francisco**
- **Problemas de Ingeniería Agrícola**
- **Temas Actuales: deforestación, cambio climático, ...**



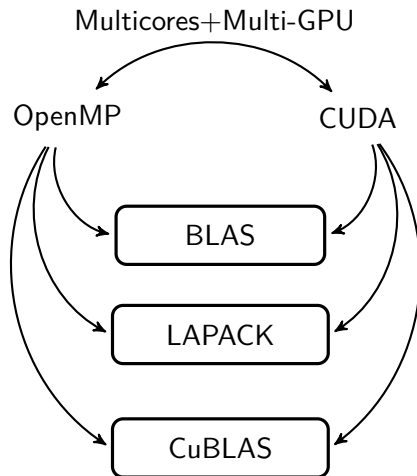
Problemas a tratar en esta tesis



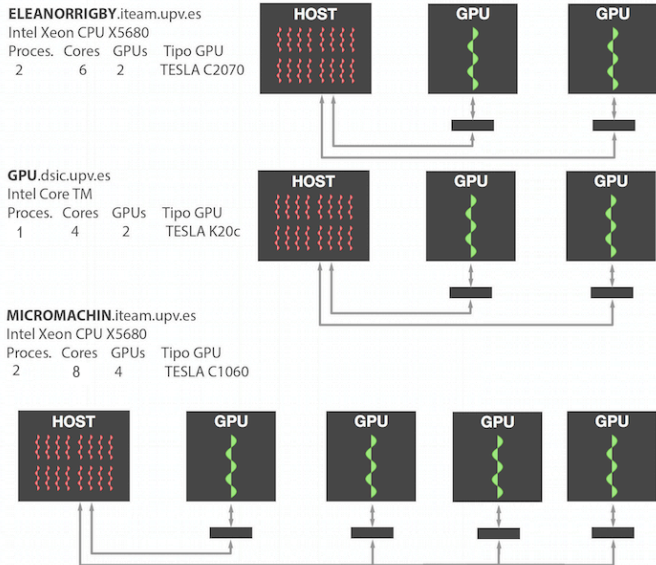
Clasificación de trabajos bibliográficos.

Referencias	Descripción	G1	G2	G3
[5, 6]	Aplicaciones	●	—	●
[17, 18, 19, 20, 21, 22, 23]	Arquitecturas	●	—	●
[24]	Heurísticas	●	—	—
[7, 8, 11, 12]	Modelos Analíticos	●	—	●
[42, 43]	Proyecto ATLAS	—	●	●
[45, 46, 47, 51]	Proyecto HeteroMPI	—	●	●
[4, 44]	Proyecto mpC	—	●	●
[49]	Proyecto FLAMINGO	—	●	●
[9, 10, 25, 26]	Modelos de Rendimiento	●	●	●
[27, 29, 31]	Herramientas	—	●	●

Herramientas de software



Herramientas de hardware



Contenido

Introducción y Contextualización de la investigación

Planteamiento, objetivos, ámbito del trabajo, problemas a tratar, estado del arte y herramientas computacionales

Problemas Aplicados a Ingeniería Agrícola

Representación del Relieve

Polinomios Matriciales

Variables Meteorológicas del Clima

Modelado Hídrico de Ríos

Autooptimización en Aplicaciones Paralelas

Conclusiones Generales y Aportaciones

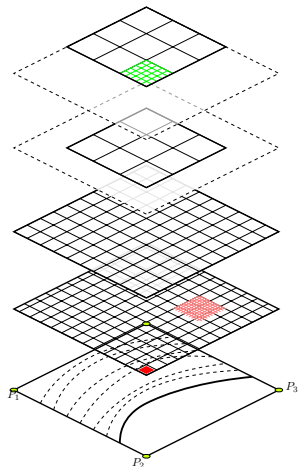
Representación del Relieve

Representación del Relieve

- El **Relieve** es un conjunto de formas complejas que accidentan la superficie del globo terráqueo.
- Nos centramos en la medición de las zonas agrícolas, especialmente en la región situada en **Valle del Rio São Francisco** en **Brasil**.
- Los factores preponderantes que han motivado la representación de **Relieve** son:
 - El mejor diseño de la zona agrícola.
 - Optimización del uso del agua.
 - Minimización de los efectos de erosión del suelo.

Representación del Relieve

- **Entrada:** Longitud, Latitud y Altitud
- **Tipo:** Matriz Densa
- **Problema Computacional:**
$$\mathbf{A} = \sum_{i=0}^m \sum_{j=0}^n x_i^\alpha y_j^\beta$$
$$\mathbf{A}\mathbf{x} = \mathbf{b}$$
- **Salida:** Polinomio de Alto Grado



Representación del Relieve

● ENTRADA

- (x, y, z) : Grupo de coordenadas geográficas representando longitud, latitud y altitud.
- r, s : Grado del Polinomio Bidimensional x y y .
- m, n : Tamaño y ancho de la superficie.

● SALIDA

- Una función de elevación que estime esos puntos.

$$z_{ij} = f(x_i, y_j) = \sum_{k=0}^r \sum_{l=0}^s c_{kl} x_i^k y_j^l$$

● MÉTODO

- Construir una matriz de coeficientes \mathbf{A} y vector de términos independientes \mathbf{b} .

$$\mathbf{A}_{kl} = \sum_{i=0}^m \sum_{j=0}^n x_i^\alpha y_j^\beta$$

$$\mathbf{b}_k = \sum_{i=0}^m \sum_{j=0}^n z_{ij} x_i^\alpha y_j^\beta$$

- Se obtiene los coeficientes que minimicen el error en función de estimación $f(x, y)$.

$$\mathbf{Ax} = \mathbf{b}$$

Representación del Relieve

● VENTAJAS

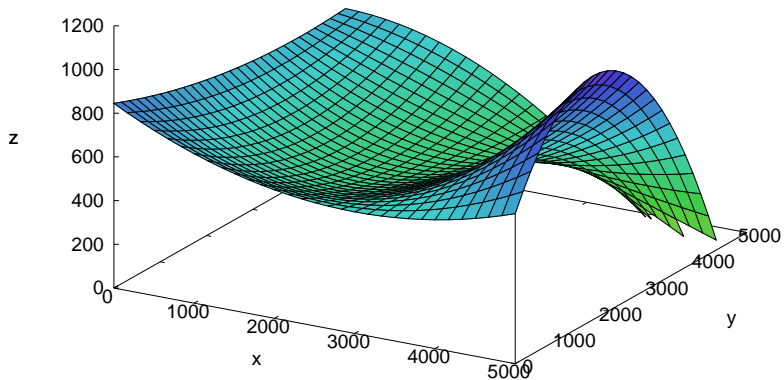
- **Reducción** significativa de la cantidad de información necesaria para la Representación del Relieve.
- A través de la Representación del Relieve es posible generar imágenes con **diferentes niveles de resolución**.

● DESVENTAJA

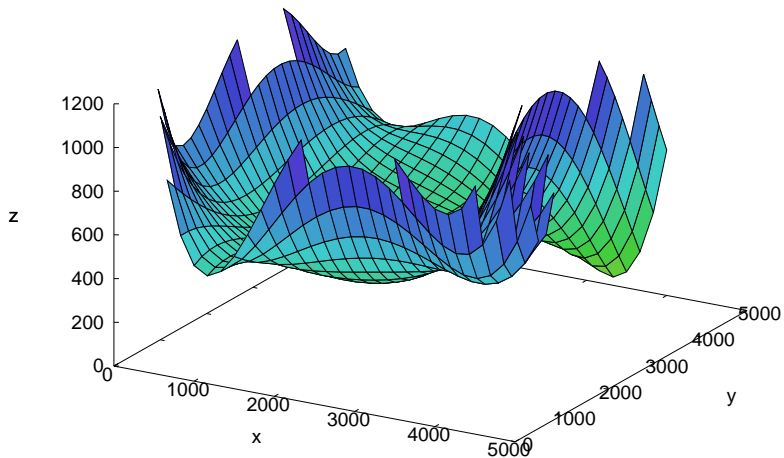
- La **complejidad** y la **potencia de cálculo** exigido en la obtención de tales polinomios es muy alto y algunas veces poco práctico.

Se requiere un **polinomio de un alto grado** para representar el **Relieve** fielmente, también exige una gran **potencia computacional** y un gran **tiempo de ejecución**.

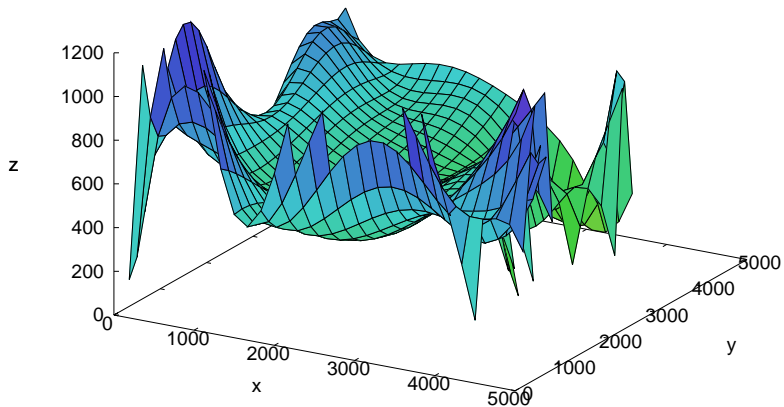
Visión 3D del Polinomio de grado 2



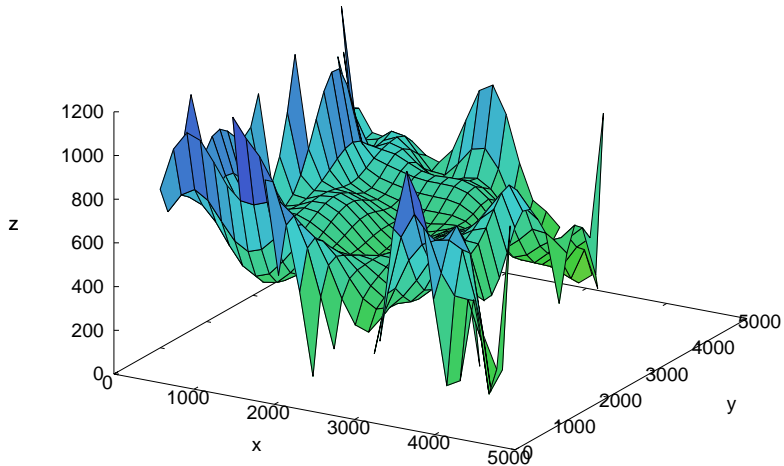
Visión 3D del Polinomio de grado 4



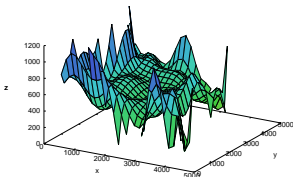
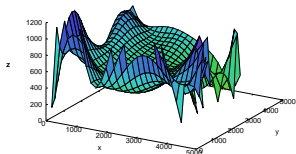
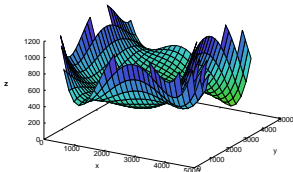
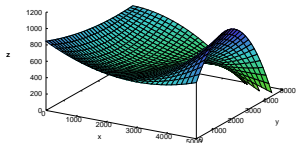
Visión 3D del Polinomio de grado 6



Visión 3D del Polinomio de grado 20



Representación del Relieve



Visión 3D de la Representación de Relieve del Valle del Río São Francisco para los grados 2, 4, 6 y 20.

Tonos de Gris del Polinomio de grado 2



Tonos de Gris del Polinomio de grado 4



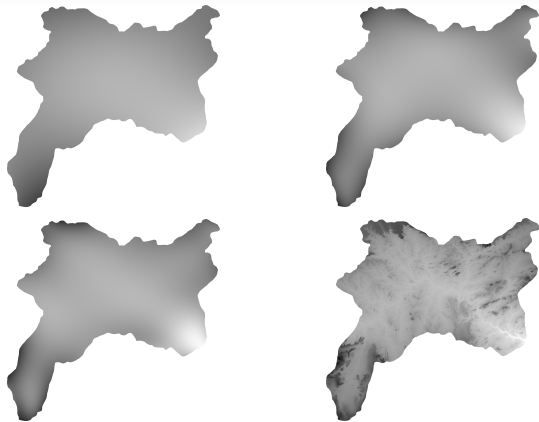
Tonos de Gris del Polinomio de grado 6



Tonos de Gris del Polinomio de grado 20



Representación del Relieve



Representación de Relieve en Tonos de Gris del Valle del Rio São Francisco para los grados polinomiales 2, 4, 6 y 20.

Representación del Relieve

Grado del Polinomio	Tiempo de Ejecución
30	2.8 días
60	25.9 días
80	2.1 meses
100	4.2 meses
150	1.2 años
225	4.1 años
300	9.9 años
375	19.5 años
450	36.9 años
500	46.5 años

Estimación del tiempo necesario en meses y/o años para obtener el resultado de la Representación del Relieve.

Representación del Relieve

Grado del Polinomio	Tiempo de Ejecución
30	2.8 días
60	25.9 días
80	2.1 meses
100	4.2 meses
150	1.2 años
225	4.1 años
300	9.9 años
375	19.5 años
450	36.9 años
500	46.5 años

Estimación del tiempo necesario en meses y/o años para obtener el resultado de la Representación del Relieve.

Distribución de la Carga de Trabajo

La construcción de la matriz **A** y del vector **b** es la parte **más costosa** del proceso global.

$$\hat{Z}_{ij}(x_i, y_j) = a_{00}x^0y^0 + a_{01}x^0y^1 + a_{02}x^0y^2 + a_{10}x^1y^0 + a_{11}x^1y^1 + a_{12}x^1y^2 + a_{20}x^2y^0 + a_{21}x^2y^1 + a_{22}x^2y^2$$

$$\begin{bmatrix}
 \sum_{i=0}^n \sum_{j=0}^n x_i^0 y_j^0 & \sum_{i=0}^n \sum_{j=0}^n x_i^0 y_j^1 & \sum_{i=0}^n \sum_{j=0}^n x_i^0 y_j^2 & \sum_{i=0}^n \sum_{j=0}^n x_i^1 y_j^0 & \sum_{i=0}^n \sum_{j=0}^n x_i^1 y_j^1 & \sum_{i=0}^n \sum_{j=0}^n x_i^1 y_j^2 & \sum_{i=0}^n \sum_{j=0}^n x_i^2 y_j^0 & \sum_{i=0}^n \sum_{j=0}^n x_i^2 y_j^1 & \sum_{i=0}^n \sum_{j=0}^n x_i^2 y_j^2 \\
 \sum_{i=0}^n \sum_{j=0}^n x_i^0 y_j^1 & \sum_{i=0}^n \sum_{j=0}^n x_i^0 y_j^2 & \sum_{i=0}^n \sum_{j=0}^n x_i^1 y_j^0 & \sum_{i=0}^n \sum_{j=0}^n x_i^1 y_j^1 & \sum_{i=0}^n \sum_{j=0}^n x_i^1 y_j^2 & \sum_{i=0}^n \sum_{j=0}^n x_i^2 y_j^0 & \sum_{i=0}^n \sum_{j=0}^n x_i^2 y_j^1 & \sum_{i=0}^n \sum_{j=0}^n x_i^2 y_j^2 \\
 \sum_{i=0}^n \sum_{j=0}^n x_i^0 y_j^2 & \sum_{i=0}^n \sum_{j=0}^n x_i^1 y_j^0 & \sum_{i=0}^n \sum_{j=0}^n x_i^1 y_j^1 & \sum_{i=0}^n \sum_{j=0}^n x_i^1 y_j^2 & \sum_{i=0}^n \sum_{j=0}^n x_i^2 y_j^0 & \sum_{i=0}^n \sum_{j=0}^n x_i^2 y_j^1 & \sum_{i=0}^n \sum_{j=0}^n x_i^2 y_j^2 \\
 \sum_{i=0}^n \sum_{j=0}^n x_i^1 y_j^0 & \sum_{i=0}^n \sum_{j=0}^n x_i^1 y_j^1 & \sum_{i=0}^n \sum_{j=0}^n x_i^1 y_j^2 & \sum_{i=0}^n \sum_{j=0}^n x_i^2 y_j^0 & \sum_{i=0}^n \sum_{j=0}^n x_i^2 y_j^1 & \sum_{i=0}^n \sum_{j=0}^n x_i^2 y_j^2 \\
 \sum_{i=0}^n \sum_{j=0}^n x_i^1 y_j^1 & \sum_{i=0}^n \sum_{j=0}^n x_i^1 y_j^2 & \sum_{i=0}^n \sum_{j=0}^n x_i^2 y_j^0 & \sum_{i=0}^n \sum_{j=0}^n x_i^2 y_j^1 & \sum_{i=0}^n \sum_{j=0}^n x_i^2 y_j^2 \\
 \sum_{i=0}^n \sum_{j=0}^n x_i^1 y_j^2 & \sum_{i=0}^n \sum_{j=0}^n x_i^2 y_j^0 & \sum_{i=0}^n \sum_{j=0}^n x_i^2 y_j^1 & \sum_{i=0}^n \sum_{j=0}^n x_i^2 y_j^2 \\
 \sum_{i=0}^n \sum_{j=0}^n x_i^2 y_j^0 & \sum_{i=0}^n \sum_{j=0}^n x_i^2 y_j^1 & \sum_{i=0}^n \sum_{j=0}^n x_i^2 y_j^2 \\
 \sum_{i=0}^n \sum_{j=0}^n x_i^2 y_j^1 & \sum_{i=0}^n \sum_{j=0}^n x_i^2 y_j^2 \\
 \sum_{i=0}^n \sum_{j=0}^n x_i^2 y_j^2
 \end{bmatrix}
 \begin{bmatrix}
 \sum_{i=0}^n \sum_{j=0}^n Z_{ij} x_i^0 y_j^0 \\
 \sum_{i=0}^n \sum_{j=0}^n Z_{ij} x_i^0 y_j^1 \\
 \sum_{i=0}^n \sum_{j=0}^n Z_{ij} x_i^0 y_j^2 \\
 \sum_{i=0}^n \sum_{j=0}^n Z_{ij} x_i^1 y_j^0 \\
 \sum_{i=0}^n \sum_{j=0}^n Z_{ij} x_i^1 y_j^1 \\
 \sum_{i=0}^n \sum_{j=0}^n Z_{ij} x_i^1 y_j^2 \\
 \sum_{i=0}^n \sum_{j=0}^n Z_{ij} x_i^2 y_j^0 \\
 \sum_{i=0}^n \sum_{j=0}^n Z_{ij} x_i^2 y_j^1 \\
 \sum_{i=0}^n \sum_{j=0}^n Z_{ij} x_i^2 y_j^2
 \end{bmatrix}
 \begin{matrix}
 a_{00} \\
 a_{01} \\
 a_{02} \\
 a_{10} \\
 a_{11} \\
 a_{12} \\
 a_{20} \\
 a_{21} \\
 a_{22}
 \end{matrix}$$

Parte del Algoritmo Secuencial de Representación del Relieve

La construcción de la matriz **A** y el vector **b** es la parte **más costosa** del proceso global.

matrixComputation para la construcción de la matriz *A*

```
for  $k = 0, \dots, (s + 1)^2$  do
{
    for  $l = 0, \dots, (s + 1)^2$  do
    {
        for  $i = 0, \dots, m \times n$  do
             $A[k + l * (s + 1)^2] + = x[i]^{(k+l)/s} * y[i]^{(k+l)\%s}$ 
        }
    }
}
```

Programación en un Entorno Heterogéneo

Paralelismo Intrínseco

- Todos los elementos de la matriz A se pueden calcular de forma simultánea.
- Cada término de la suma puede ser calculado de forma independiente y concurrente.

Tareas Particionadas

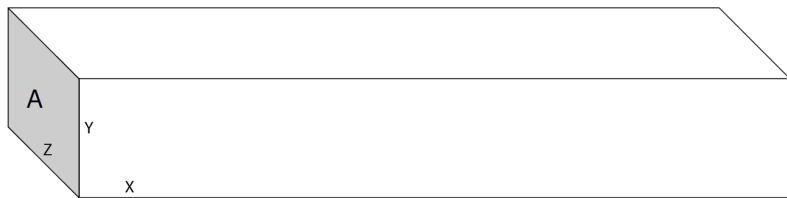
- Se reparte la suma en partes o tareas de distintos tamaños.
- La cantidad de carga de trabajo que debe ser enviado a cada recurso para que se reduzca al mínimo el tiempo total.

Distribución de la Carga de Trabajo

- La Distribución de la Carga de Trabajo es hecha de manera **equilibrada**.
- Cada operación es un cálculo de un término de la suma.
- Los elementos de la matriz **A** se almacenan en la superficie izquierda sombreada.

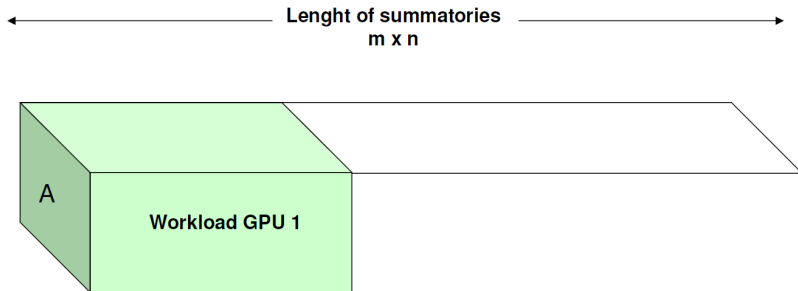
← Length of summatories →

$$A_{kl} = \sum_{i=0}^m \sum_{j=0}^n x_i^{k/(s+1)+l/(s+1)} y_j^{k\%(r+1)+l\%(r+1)}$$



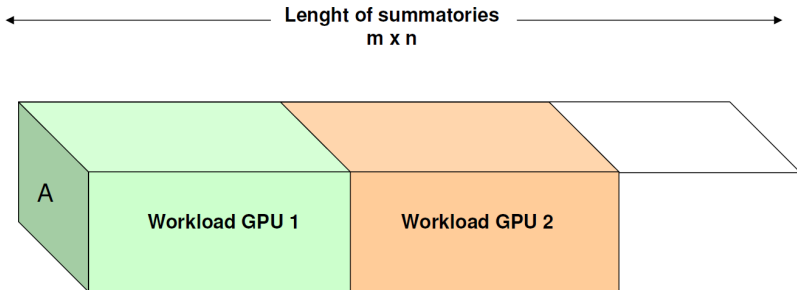
Distribución de la Carga de Trabajo

La aplicación permite predecir, la cantidad de **carga de trabajo** que debe ser enviado a cada uno de los recursos con el objetivo de minimizar el tiempo total de ejecución.



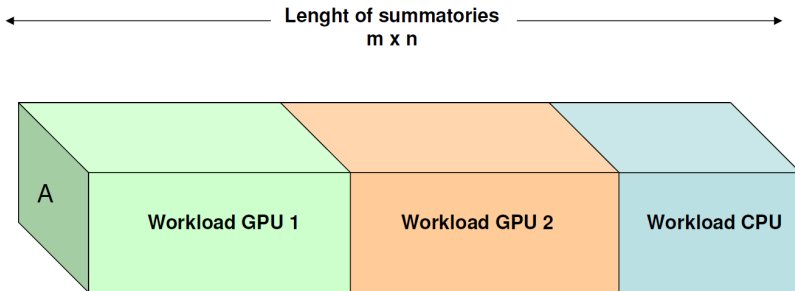
Distribución de la Carga de Trabajo

La aplicación permite predecir, la cantidad de **carga de trabajo** que debe ser enviado a cada uno de los recursos con el objetivo de minimizar el tiempo total de ejecución.

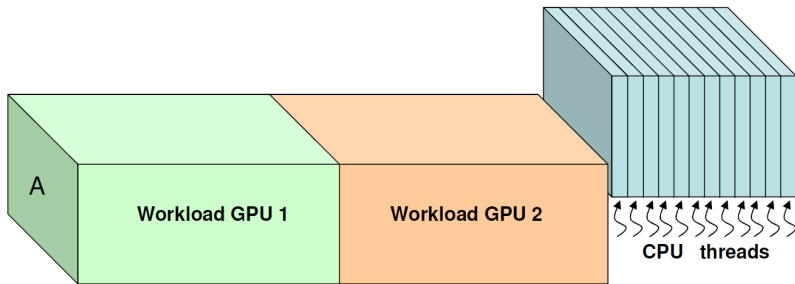


Distribución de la Carga de Trabajo

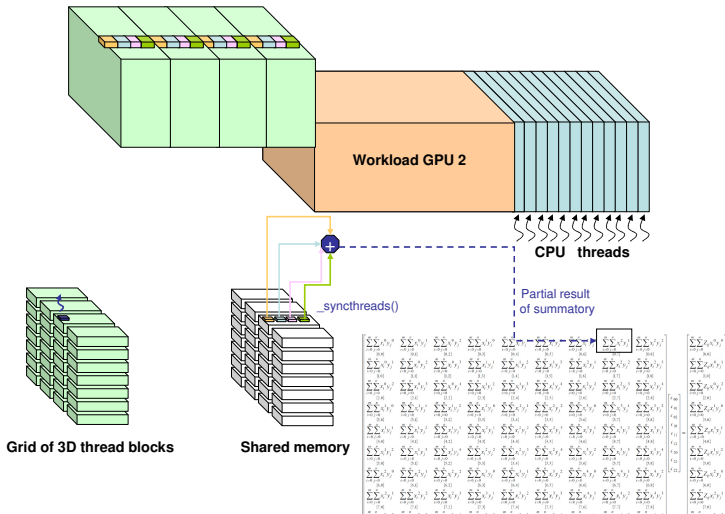
La aplicación permite predecir, la cantidad de **carga de trabajo** que debe ser enviado a cada uno de los recursos con el objetivo de minimizar el tiempo total de ejecución.



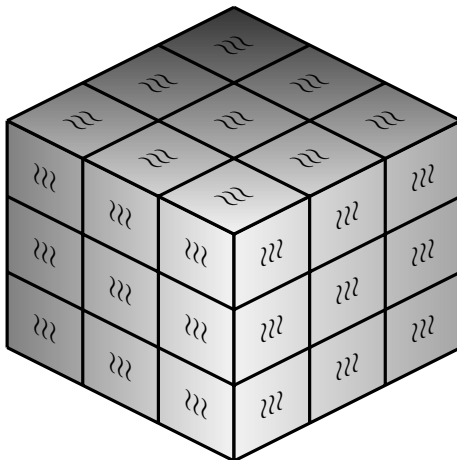
Paralelismo en los cores



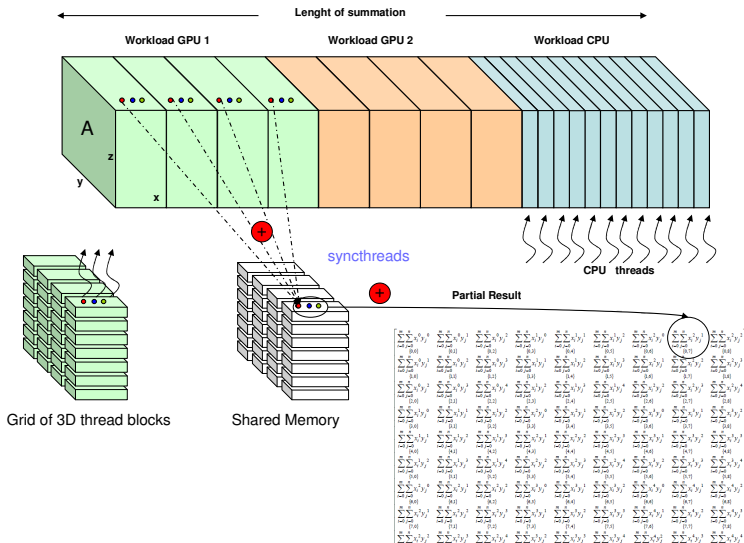
Paralelismo en los cores



Paralelismo en los cores



Paralelismo en los cores



Resultados Experimentales

Caracterización de la Plataforma de Ejecución

Nombre	ELEANORRIGBY
Procesador	Intel®Xeon CPU X5680
Memoria	96GB
Clock	3.33GHz
Número de Procesadores	2
Cores por Procesador	6
GPU	NVIDIA TESLA C2070
Número de GPUs	2
CUDA cores por GPU	448
GPU Memory por GPU	6GB GRR3
Versión CUDA	4.0

Gráfica del Tiempo de Ejecución

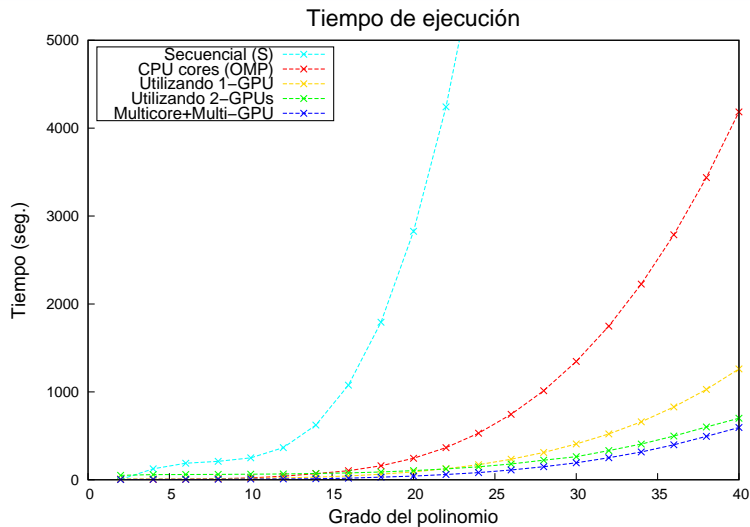


Tabla Speedup

Tabla con los *Speedups* variando el tamaño del problema (grado del polinomio).

Grado	OMP	1-GPU	2-GPUs	híbrido
8	6.85	6.79	6.26	7.69
12	9.22	18.07	21.13	30.40
16	12.18	28.93	43.05	50.70
20	12.59	35.38	60.84	64.67
24	11.85	36.41	68.52	71.76
28	11.94	36.65	69.50	74.60
32	11.72	36.77	69.58	75.07
36	12.45	37.32	69.62	76.50
40	13.00	37.12	70.27	78.00

● Conclusiones de la Representación del Relieve

- El **paralelismo híbrido** (múltiples niveles, diferentes tipos de paralelismo, diferentes paradigmas ...) presenta dificultades adicionales.
- El **alto rendimiento** logrado con nuestra aplicación híbrida permite reducir el tiempo para obtener un polinomio de alto grado que representa con fidelidad el relieve.

Contenido

Introducción y Contextualización de la investigación

Planteamiento, objetivos, ámbito del trabajo, problemas a tratar, estado del arte y herramientas computacionales

Problemas Aplicados a Ingeniería Agrícola

Representación del Relieve

Polinomios Matriciales

Variables Meteorológicas del Clima

Modelado Hídrico de Ríos

Autooptimización en Aplicaciones Paralelas

Conclusiones Generales y Aportaciones

Polinomios Matriciales

Polinomios Matriciales

- La idea de los **Polinomios Matriciales** es la reconstrucción de una superficie de la que se posee almacenada la información en forma de polinomio bidimensional.
- A través de polinomios matriciales se pueden describir las superficies espaciales de manera **más reducida** y, por lo tanto, ahorrando cómputo.
- Su estudio ofrece la **oportunidad** de:
 - Construir algoritmos para cálculos de funciones matriciales.
 - Desarrollar Interfaces MATLAB.
 - Ofrecer a usuarios no expertos herramientas más apropiadas para solucionar un problema.

Definición de Polinomios Matriciales

$$P = \sum_{i=0}^d \alpha_i X^i = \alpha_d X^d + \alpha_{d-1} X^{d-1} + \dots + \alpha_1 X + \alpha_0 I,$$

para $\bar{\alpha} = [\alpha_i]_{i=0,\dots,d}$ y $X, I \in \mathcal{R}^{n \times n}$, siendo I la matriz identidad.

Estrategias de resolución del los Polinomios Matriciales

- 1 Construcción de un **Algoritmo Sencillo**, que resuelve la ecuación anterior. La estrategia consiste en el cálculo y almacenamiento de los términos pares y impares del polinomio matricial.
- 2 Construcción de un **Algoritmo Recursivo utilizando la técnica de encajonamiento**.

Un algoritmo sencillo para evaluar Polinomios Matriciales

Algoritmo para la evaluación de un polinomio matricial.

```
1: procedure EVALUATE(  $n, X, d, \bar{\alpha}$ )
2:    $P \leftarrow \alpha_0 I$ 
3:    $P \leftarrow P + \alpha_1 X$ 
4:    $B \leftarrow X$ 
5:   for  $i \leftarrow 2, d$  do
6:      $A \leftarrow B$ 
7:      $B \leftarrow A \cdot X$ 
8:      $P \leftarrow P + \alpha_i B$ 
9:   end for
10: end procedure
```

Un algoritmo sencillo para evaluar Polinomios Matriciales (Multi-GPU)

Algoritmo para el cálculo de los índices de X utilizando 2-GPUs.

```
1: function COMPUTE_POWERS(  $d, X$  ) return ( $A, m$ )
2:   #pragma omp parallel for
3:   for  $g \leftarrow 0, 1$  do
4:      $m = d/2 + g \cdot \text{mod}(d, 2)$ 
5:      $A(0) \leftarrow X$ 
6:      $A(1) \leftarrow X \cdot A(0)$  ▷  $A(1) = X^2$ 
7:     if  $g = 0$  then
8:        $A(0) \leftarrow A(1)$  ▷  $A(0) = X^2$  if  $g = 0$ 
9:     end if
10:     $A(2) \leftarrow A(0) \cdot A(1)$  ▷  $A(2) = \begin{cases} X^4 & \text{if } g = 0 \\ X^3 & \text{if } g = 1 \end{cases}$ 
11:    for  $i \leftarrow 3, m$  do
12:       $A(i) \leftarrow A(1) \cdot A(i-1)$  ▷  $A(i) = X^2 \cdot A(i-1)$ 
13:    end for
14:  end for
15: end function
```

Polinomios Matriciales

Las **potencias** de la matriz X están almacenadas en un vector o array de matrices A ,

$$A = [A(i)]_{i=0,\dots,m}, A(i) \in \mathcal{R}^{n \times n},$$

para $m = \frac{d}{2} + g \cdot \text{mod}(d, 2)$, donde $g \in \{0, 1\}$ es el número de recursos computacionales disponibles. El valor de estos vectores para cada GPU después de la ejecución del Algoritmo 2 es

$$A = \begin{cases} \left(\begin{array}{ccccc} X^2 & X^4 & X^6 & X^8 & \dots \end{array} \right) & \text{if GPU0} \\ \left(\begin{array}{ccccc} X^1 & X^3 & X^5 & X^7 & \dots \end{array} \right) & \text{if GPU1} \end{cases}$$

Un algoritmo paralelo eficiente utilizando *encajonamiento*

$$\begin{aligned}P &= \sum_{i=0}^{14} \alpha_i X^i \\&= \alpha_{14} X^{14} + \alpha_{13} X^{13} + \alpha_{12} X^{12} + \alpha_{11} X^{11} + \alpha_{10} X^{10} + \alpha_9 X^9 + \alpha_8 X^8 \\&+ \alpha_7 X^7 + \alpha_6 X^6 + \alpha_5 X^5 + \alpha_4 X^4 + \alpha_3 X^3 + \alpha_2 X^2 + \alpha_1 X + \alpha_0 I\end{aligned}$$

$$\begin{aligned}&= X^{12} (\alpha_{14} X^2 + \alpha_{13} X + \alpha_{12} I) + X^8 (\alpha_{11} X^3 + \alpha_{10} X^2 + \alpha_9 X + \alpha_8 I) \\&+ X^4 (\alpha_7 X^3 + \alpha_6 X^2 + \alpha_5 X + \alpha_4 I) + (\alpha_3 X^3 + \alpha_2 X^2 + \alpha_1 X + \alpha_0 I)\end{aligned}$$

$$Q^q(\bar{\alpha}) = Q^q(\bar{\alpha}, X) = \alpha_q X^q + \alpha_{q-1} X^{q-1} + \dots + \alpha_1 X + \alpha_0 I$$

Un algoritmo paralelo eficiente utilizando *encajonamiento*

$$\begin{aligned} P &= X^{12} Q^2(\bar{\alpha}_{14:12}) + X^8 Q^3(\bar{\alpha}_{11:8}) + X^4 Q^3(\bar{\alpha}_{7:4}) + Q^3(\bar{\alpha}_{3:0}) \\ &= X^4 (X^4 (X^4 (Q^2(\bar{\alpha}_{14:12})) + Q^3(\bar{\alpha}_{11:8}))) + Q^3(\bar{\alpha}_{7:4}) + Q^3(\bar{\alpha}_{3:0}) \end{aligned}$$

$$A = [X^{i+1}]_{i=0,\dots,b} = \left(X \quad X^2 \quad X^3 \quad \dots \quad X^{b-1} \quad X^b \quad X^{b+1} \right)$$

$$P_i \leftarrow X^{b+1} P_{b+i+1} + Q^b(\bar{\alpha}_{b+i:i}) = X^{b+1} Q_1 + Q_2 ,$$

para $i = 0, b+1, 2(b+1), 3(b+1), \dots$

Un algoritmo paralelo eficiente utilizando *encajonamiento*

$$\begin{aligned} P &= X^{12}Q^2(\bar{\alpha}_{14:12}) + X^8Q^3(\bar{\alpha}_{11:8}) + X^4Q^3(\bar{\alpha}_{7:4}) + Q^3(\bar{\alpha}_{3:0}) \\ &= X^4(X^4(X^4(Q^2(\bar{\alpha}_{14:12}))) + Q^3(\bar{\alpha}_{11:8})) + Q^3(\bar{\alpha}_{7:4}) + Q^3(\bar{\alpha}_{3:0}) \end{aligned}$$

$$A = [X^{i+1}]_{i=0,\dots,b} = \begin{pmatrix} X & X^2 & X^3 & \dots & X^{b-1} & X^b & X^{b+1} \end{pmatrix}$$

$$P_i \leftarrow X^{b+1}P_{b+i+1} + Q^b(\bar{\alpha}_{b+i:i}) = X^{b+1}Q_1 + Q_2,$$

para $i = 0, b+1, 2(b+1), 3(b+1), \dots$

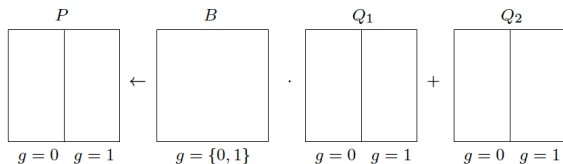
Recursividad + encajonamiento

$$X^{p-1} = \begin{pmatrix} X_1 & X_2 \end{pmatrix},$$

$$X^p = \begin{pmatrix} Y_1 & Y_2 \end{pmatrix},$$

donde $X_1, Y_1, X_2, Y_2 \in \mathbb{R}^{n \times (n/2)}$ (se asume por simplicidad que n es par), entonces

$$X^p = X \cdot X^{p-1} = X \cdot \begin{pmatrix} X_1 & X_2 \end{pmatrix} = \begin{pmatrix} X \cdot X_1 & X \cdot X_2 \end{pmatrix} = \begin{pmatrix} Y_1 & Y_2 \end{pmatrix},$$



Multiplicación de 2 matrices en 2 GPUs.

Recursividad + encajonamiento

Algoritmo recursivo para la evaluación de polinomios matriciales utilizando *encajonamiento*.

```
1: function ENCAJONAMIENTO(  $n, d, b, i, \bar{\alpha}, A$  ) return  $P$ 
2:    $q \leftarrow d - i$ 
3:   if  $q \leq b + 1$  then
4:      $P \leftarrow$  EVALUATE(  $n, A, d, \bar{\alpha}_{q+i:i}$  )
5:   else
6:      $Q_1 \leftarrow$  ENCAJONAMIENTO(  $n, d, b, i + b + 1, \bar{\alpha}, A$  )
7:      $Q_2 \leftarrow$  EVALUATE(  $n, A, d, \bar{\alpha}_{q+i:i}$  )
8:      $P \leftarrow A(q) \cdot Q_1 + Q_2$ 
9:   end if
10: end function
```

Algoritmo recursivo utilizando *encajonamiento* para 2-GPUs

Algoritmo recursivo para la evaluación de polinomios matriciales con 2-GPUs utilizando *encajonamiento*.

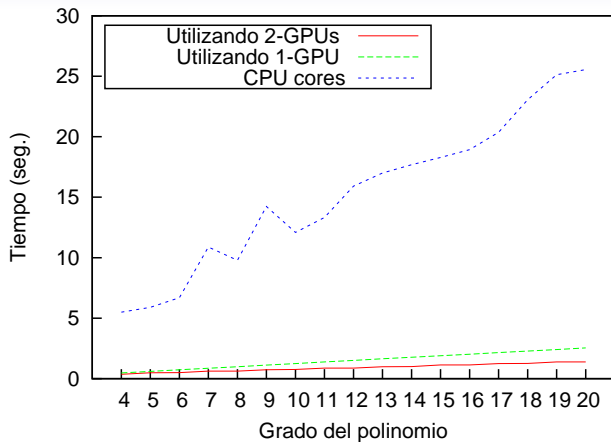
```
1: function ENCAJONAMIENTO2(  $d, b, i, \bar{\alpha}, A, B$  ) return  $P$ 
2:   #pragma omp parallel for
3:   for  $g \leftarrow 0, 1$  do
4:      $q \leftarrow d - i$ 
5:     if  $q < b$  then
6:        $P \leftarrow \text{EVALUATE}( q, A, d, \bar{\alpha}_{q+i:i} )$ 
7:     else
8:        $Q_1 \leftarrow \text{ENCAJONAMIENTO2}( d, b, i + b + 1, \bar{\alpha}, A, B )$ 
9:        $Q_2 \leftarrow \text{EVALUATE}( q, A, d, \bar{\alpha}_{q+i:i} )$ 
10:       $P \leftarrow B \cdot Q_1 + Q_2$ 
11:    end if
12:  end for
13: end function
```

Resultados Experimentales

Caracterización de la Plataforma de Ejecución

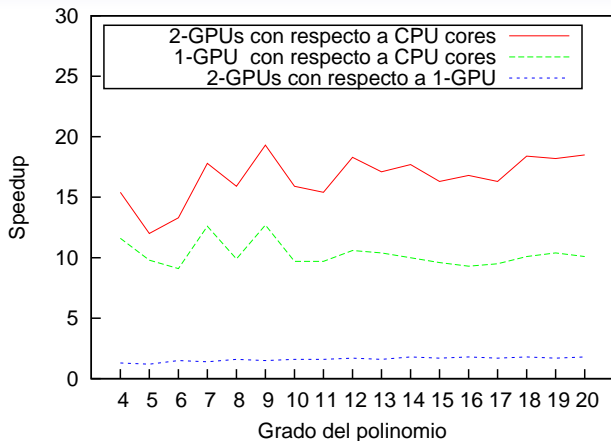
	GPU
Nombre	
Procesador	Intel®Core (TM)
Memoria	16GB
Clock	3.60GHz
Número de Procesadores	1
Cores por Procesador	4
GPU	NVIDIA TESLA K20c
Número de GPUs	2
CUDA cores por GPU	2496
GPU Memory por GPU	5GB GRR3
Versión CUDA	4.0

Algoritmo Sencillo



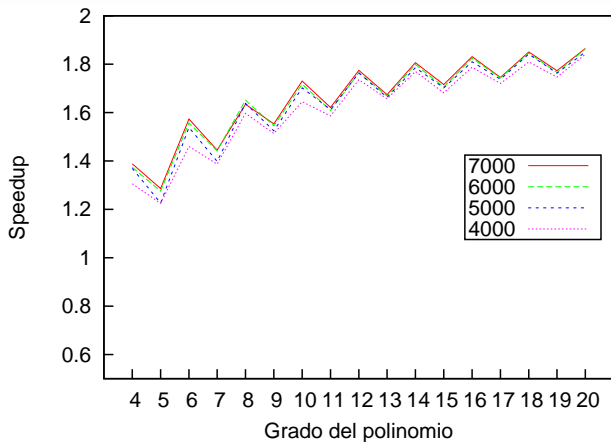
Tiempo de ejecución para evaluar un polinomio matricial con matriz de tamaño $n = 4000$ y variando el grado.

Algoritmo Sencillo



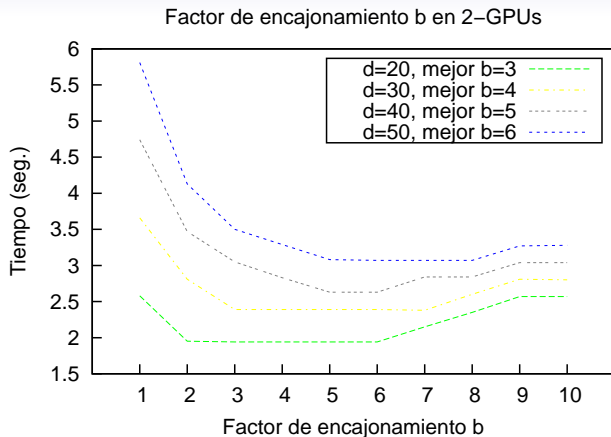
Speedup para evaluar un polinomio matricial con matriz de tamaño $n = 4000$ y variando el grado.

Algoritmo Sencillo



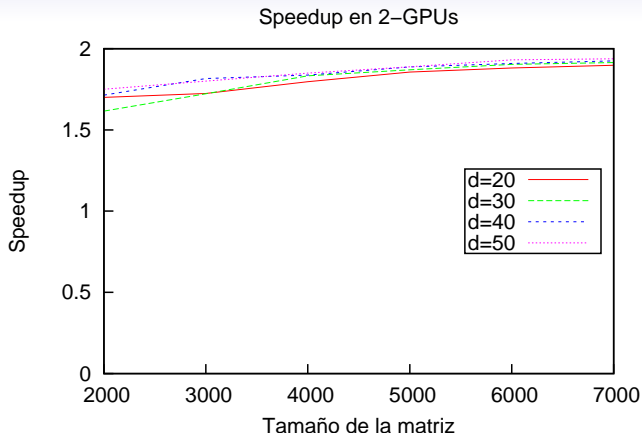
Speedup para la evaluación de polinomios matriciales en 2-GPUs con respecto a 1-GPU.

Algoritmo Recursivo + *encajonamiento*



Gráficas de tiempo de ejecución para la evaluación de un polinomio con tamaño de matriz $n = 6000$ para distintos grados de polinomio en función del factor de *encajonamiento* b en 2-GPU.

Algoritmo Recursivo + *encajonamiento*



Gráficas con el *speedup* obtenido por 2-GPU con respecto a 1-GPU para la evaluación de un polinomio con tamaño fijo de matriz $n = 6000$ variando el tamaño de la matriz y en función del grado del polinomio, utilizando un factor de encajonamiento óptimo.

- **Conclusiones de los Polinomios Matriciales - Algoritmo Recursivo + *encajonamiento***

- Nuestra motivación surge de la **ausencia** de software y/o contribuciones que traten este problema y de su utilidad en el contexto descrito en el capítulo anterior.
- La propuesta funciona bien en el caso de que el grado del polinomio sea alto y se utilice *encajonamiento*.
- Se aporta la propuesta de un modelo diferente de lo usual y sencillo con objeto de gestionar esta concurrencia. Hemos optado por un modelo de programación SPMD (*Single Program, Multiple Data*) para implementar el programa en GPU.

Contenido

Introducción y Contextualización de la investigación

Planteamiento, objetivos, ámbito del trabajo, problemas a tratar, estado del arte y herramientas computacionales

Problemas Aplicados a Ingeniería Agrícola

Representación del Relieve

Polinomios Matriciales

Variables Meteorológicas del Clima

Modelado Hídrico de Ríos

Autooptimización en Aplicaciones Paralelas

Conclusiones Generales y Aportaciones

Variables Meteorológicas del Clima

Variables Meteorológicas del Clima

- La idea es representar **variables meteorológicas del clima** en el área del Valle del Río São Francisco.
- Estas variables son conocidas gracias a **sensores** estratégicamente situados a lo largo y ancho del área bajo estudio. Sin embargo, no es posible conocer el valor de estas variables en cualquier punto por este medio debido a la poca densidad del número de sensores.
- Mediante un **método de interpolación espacial** basado en el inverso de la potencia de la distancia es posible inferir potencialmente el valor de una de estas variables en cualquier punto del terreno.
- Este método puede necesitar de una **gran capacidad computacional**, siendo además crítico el tiempo de respuesta.

Variables Meteorológicas del Clima

En este escenario, el objetivo general de esta solución consiste en **acelerar los cálculos necesarios** para realizar estas simulaciones climáticas basadas en la representación de variables meteorológicas por el método de interpolación espacial del inverso de la potencia de la distancia.

Variables Meteorológicas del Clima

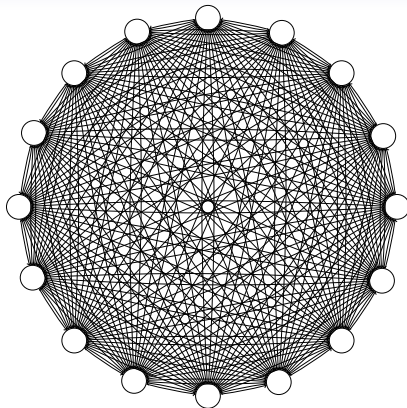
El método aplicado de **ponderación espacial del Inverso de la Potencia de la Distancia (IPD)** es un estimador de carácter determinista, con el cual se obtiene un valor interpolado a partir de una combinación lineal de los valores de los puntos de muestreo.

$$f_e(r) = \frac{\sum_{i=1}^n d(r, r_i)^{-p} (f_m(r_i))}{\sum_{i=1}^n d(r, r_i)^{-p}}$$

Descripción de los parámetros de la fórmula de la interpolación espacial del inverso de la potencia de la distancia (IPD).

Símbolo	Descripción
$f_e(r)$	Valor estimado de f_e en el vector posición r
$f_m(r_i)$	Valor medido de f_m en el vector posición r_i
n	Número total de puntos conocidos y utilizados
$d(r, r_i)$	Distancia euclidiana entre los vectores r y r_i
p	Parámetro de potencia

Variables Meteorológicas del Clima



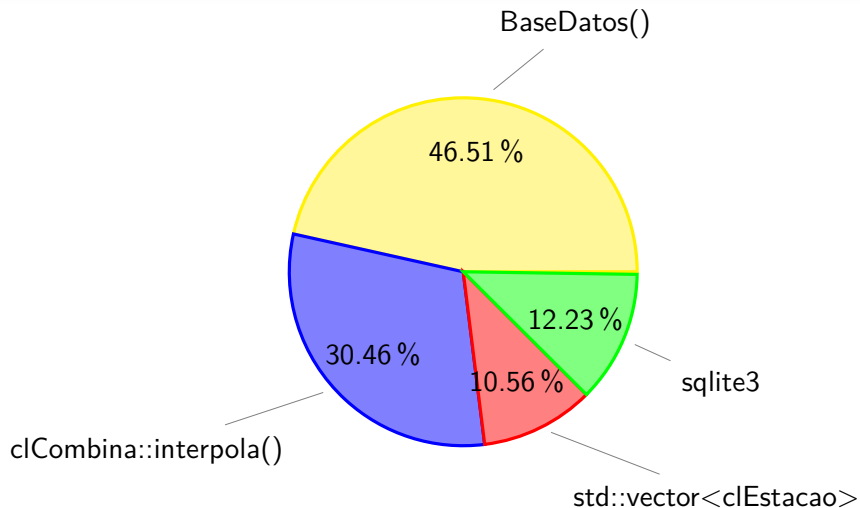
Esquema de la representación de variables meteorológicas del clima, considerando los círculos las estaciones meteorológicas y los cruces de las rectas las combinaciones lineales de los valores de los puntos de muestreo.

Variables Meteorológicas del Clima

Flat profile:

%	cumulative	self	
time	seconds	seconds	name
46.51	1478.42	1546.34	BaseDatos()
30.46	2814.31	2814.31	clCombina::interpola()
10.56	5510.12	2695.81	std::vector<clEstacao>
12.23	7114.38	1604.26	sqlite3VdbeExec
...			

Variables Meteorológicas del Clima



Resultados Experimentales

Caracterización de la Plataforma de Ejecución

Nombre	MICROMACHIN
Procesador	Intel® Xeon CPU E5530
Memoria	48GB
Clock	2.40GHz
Número de Procesadores	2
Cores por Procesador	8
GPU	NVIDIA TESLA C1060
Número de GPUs	4
CUDA cores por GPU	240
GPU Memory por GPU	8GB GRR3
Versión CUDA	4.0

Variables Meteorológicas del Clima

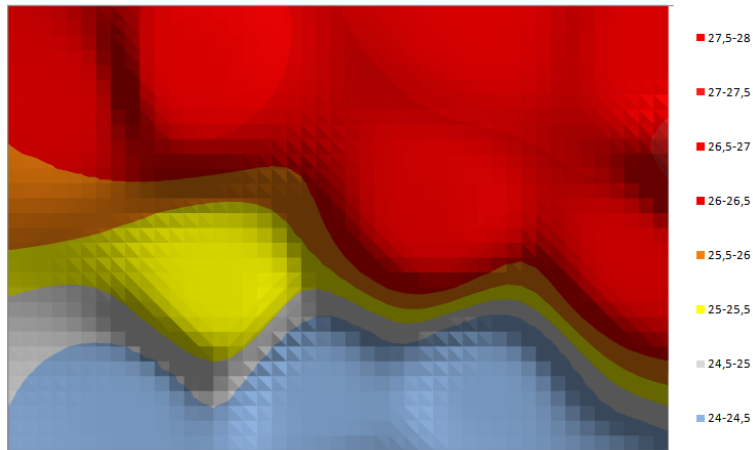
Tabla con el EAP y la p para la representación de la temperatura.

Períodos	EAP	p
Invierno	1.16	4.6
Primavera	0.94	4.8
Verano	0.97	3.3
Otoño	0.94	3.6

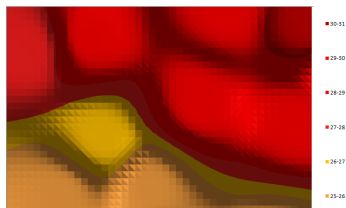
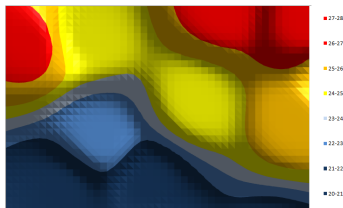
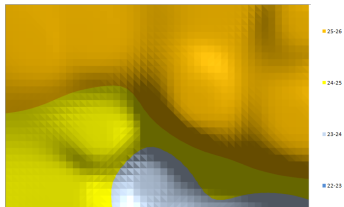
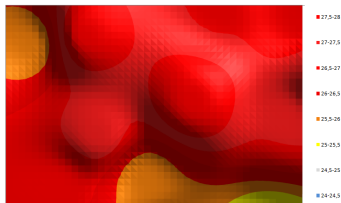
Tabla con los datos de las estaciones meteorológicas: Periodos del año, temperaturas en grados Celsius ($^{\circ}\text{C}$): mínima (T_{Min}), promedio (T_{Pro}), máxima (T_{Max}), varianza (σ^2) y la cantidad de datos experimentados (n).

Períodos	T_{Min}	T_{Pro}	T_{Max}	σ^2	n
Invierno	17.6	23.6	30.7	2.47	13 millones
Primavera	21.6	27.4	33.1	2.25	13 millones
Verano	21.4	26.7	30.9	1.65	13 millones
Otoño	19.9	24.9	31.0	1.91	13 millones

Mapa de calor de datos anuales



Mapa de calor para datos por estación del año



Tiempos de Ejecución Secuencial

Análisis comparativo de los tiempos de ejecución del algoritmo secuencial (en segundos).

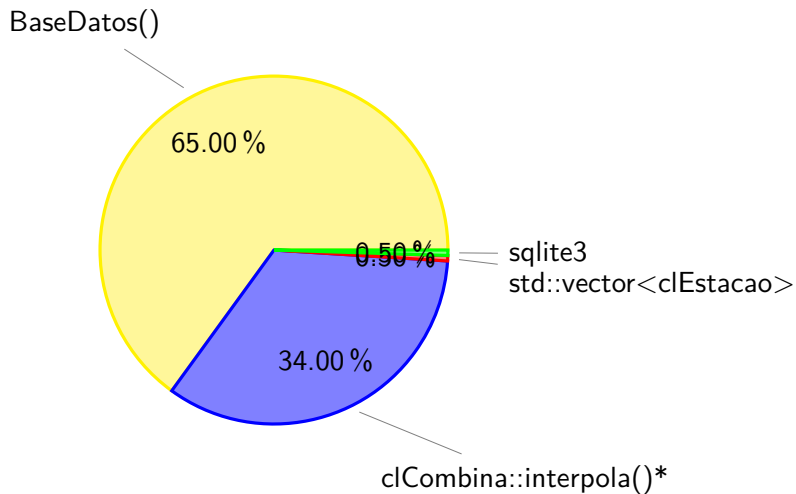
Estaciones	Aceso Base Datos	Interpolación	Computación Total
4	3.53	1.24	4.77
6	18.07	9.23	27.30
8	90.54	62.22	152.76
10	434.60	360.63	795.27
12	988.12	1996.19	2984.31
14	4472.69	21243.02	25,715.71
16	12004.17	63424.38	75428.55
18	81498.14	122191.27	203689.40
4	73.99%	26.00%	100.00%
6	66.19%	33.81%	100.00%
8	59.27%	40.73%	100.00%
10	54.65%	45.35%	100.00%
12	33.11%	66.89%	100.00%
14	17.39%	82.61%	100.00%
16	15.91%	84.08%	100.00%
18	40.01%	59.99%	100.00%

Tiempos de Ejecución Paralelo

Análisis comparativo de los tiempos de ejecución para la versión multicore+multi-GPU (en segundos).

Estaciones	Aceso Base Datos	Interpolación	Computación Total
4	2.47	5.44	7.91
6	8.35	5.25	13.60
8	38.33	20.12	59.87
10	187.34	73.33	263.49
12	284.83	157.81	442.58
14	912.75	1250.66	2154.24
16	4283.81	2458.08	6742.75
18	14498.81	7723.21	22222.84
4	46.27%	54.00%	100.00%
6	58.78%	42.09%	100.00%
8	50.97%	48.81%	100.00%
10	66.63%	33.36%	100.00%
12	61.23%	34.31%	100.00%
14	58.50%	41.50%	100.00%
16	66.72%	33.02%	100.00%
18	65.36%	34.64%	100.00%

Variables Meteorológicas del Clima



Speedups

Análisis comparativo del *Speedup* para las partes: BaseDatos, Interpolación y Computación.

Estaciones	Aceso Base Datos	Interpolación	Computación Total
18	2.52	10.23	11.27
20	2.38	10.92	12.81
22	3.48	12.64	14.39
24	4.90	11.08	14.72

● Conclusiones de las Variables Meteorológicas del Clima

- Hemos observado que el problema tiene un **grado de paralelismo intrínseco** importante, cosa que se ha aprovechado para desarrollar un algoritmo paralelo capaz de aprovechar los cores existentes en el *host*.
- El diseño completo de la **aplicación es híbrido**, es decir, utiliza todos los recursos computacionales al alcance: cores CPU y GPUs.
- La optimización de la parte computacional del problema ha dejado al descubierto, sin embargo, otro problema subyacente que no se ha podido resolver convenientemente en esta tesis y concierne al tiempo necesario de acceso a la **base de datos** (tiempo **BaseDatos**).

Contenido

Introducción y Contextualización de la investigación

Planteamiento, objetivos, ámbito del trabajo, problemas a tratar, estado del arte y herramientas computacionales

Problemas Aplicados a Ingeniería Agrícola

Representación del Relieve

Polinomios Matriciales

Variables Meteorológicas del Clima

Modelado Hídrico de Ríos

Autooptimización en Aplicaciones Paralelas

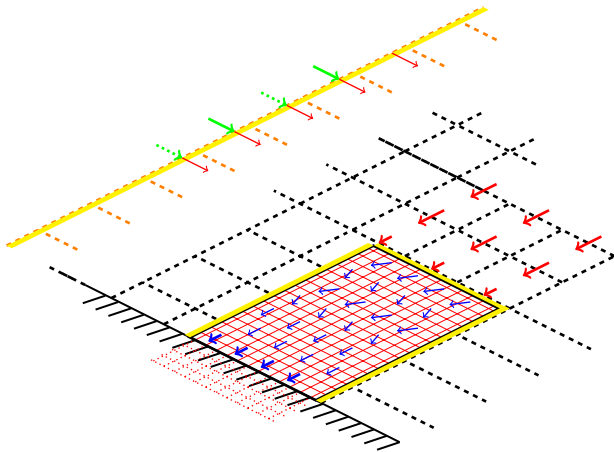
Conclusiones Generales y Aportaciones

Modelado Hídrico de Ríos

1 Cauces Hídricos en **Brasil**

- Fuertes **inundaciones** o **sequías** persistentes.
- Fenómenos que provocan un **impacto socio-económico**.
- Se estudia el comportamiento de los **acuíferos**.
- Estudio de **modelos computacionales** hídricos de ríos.
- Predecir, entender y controlar los **procesos físicos**.
- Aplicar **técnicas de computación paralela** que permitan simular los modelos hídricos y acelerar su ejecución.

Modelado Hídrico de Ríos



Flujo unidimensional en función de la velocidad del agua y de la profundidad del Río São Francisco. Variable hídrica: **escorrentía total**.

Modelado Hídrico de Ríos

El **modelo matemático hídrico** surge a partir de las ecuaciones de flujo unidimensional en función de la velocidad del agua (v) y de la profundidad del río (y):

$$b \frac{\partial y}{\partial t} + A \frac{\partial v}{\partial x} + v \cdot b \frac{\partial y}{\partial x} = 0 ,$$
$$\frac{1}{g} \frac{\partial v}{\partial t} + \frac{v}{g} \frac{\partial v}{\partial x} + \frac{\partial y}{\partial x} = (S_o - S_f) \quad (1)$$

donde g es la aceleración de la gravedad, x es la distancia longitudinal, t es el tiempo, S es la escorrentía de la línea de roce, y b el ancho del área estudiada.

Modelado Hídrico de Ríos

La derivada total de las variables dependientes es:

$$\begin{aligned} dy &= \frac{\partial y}{\partial x} dx + \frac{\partial y}{\partial t} dt, \\ dv &= \frac{\partial v}{\partial x} dx + \frac{\partial v}{\partial t} dt. \end{aligned} \tag{2}$$

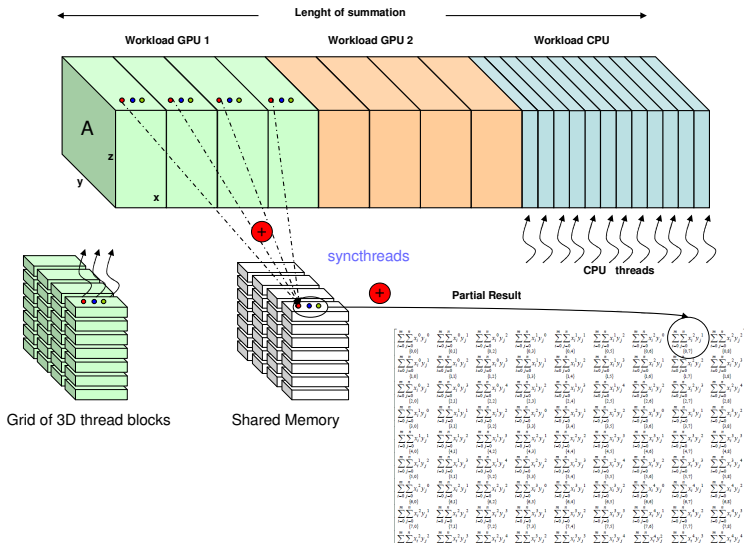
Los sistemas de la Ecuación 1 y 2 pueden ser puestos en formato matricial, teniendo como variables las derivadas parciales dependientes:

$$\begin{bmatrix} v \cdot b & b & A & 0 \\ 1 & 0 & v & 1 \\ dx & dt & 0 & 0 \\ 0 & 0 & dx & dt \end{bmatrix} \begin{bmatrix} \frac{\partial y}{\partial x} \\ \frac{\partial y}{\partial t} \\ \frac{\partial v}{\partial x} \\ \frac{\partial v}{\partial t} \end{bmatrix} = \begin{bmatrix} 0 \\ S_o - S_f \\ dy \\ dv \end{bmatrix} .$$

Parte del Código Secuencial

```
1: for  $r \rightarrow Mx, r++$  do  
2:   for  $c \rightarrow My, c++$  do  
3:      $d[0] \leftarrow d[4] \leftarrow dx$   
4:      $d[1] \leftarrow d[3] \leftarrow d[5] \leftarrow d[7] \leftarrow \text{sqrt}(dx * dx + dy * dy)$   
5:      $d[2] = d[6] \leftarrow dy$   
6:  
7:      $a[0] \leftarrow \text{atan}((Q[(r+1) * Mx + c].z - Q[r * My + c].z) / d[0])$   
8:      $a[1] \leftarrow \text{atan}((Q[(r+1) * Mx + (c+1)].z - Q[r * My + c].z) / d[1])$   
9:      $a[2] \leftarrow \text{atan}((Q[r * Mx + (c+1)].z - Q[r * My + c].z) / d[2])$   
10:     $a[3] \leftarrow \text{atan}((Q[(r-1) * Mx + (c+1)].z - Q[r * My + c].z) / d[3])$   
11:     $a[4] \leftarrow \text{atan}((Q[(r-1) * Mx + c].z - Q[r * My + c].z) / d[4])$   
12:     $a[5] \leftarrow \text{atan}((Q[(r-1) * Mx + (c-1)].z - Q[r * My + c].z) / d[5])$   
13:     $a[6] \leftarrow \text{atan}((Q[r * Mx + (c-1)].z - Q[r * My + c].z) / d[6])$   
14:     $a[7] \leftarrow \text{atan}((Q[(r+1) * Mx + (c-1)].z - Q[r * My + c].z) / d[7])$   
15:   end for  
16: end for
```

Paralelismo en los cores

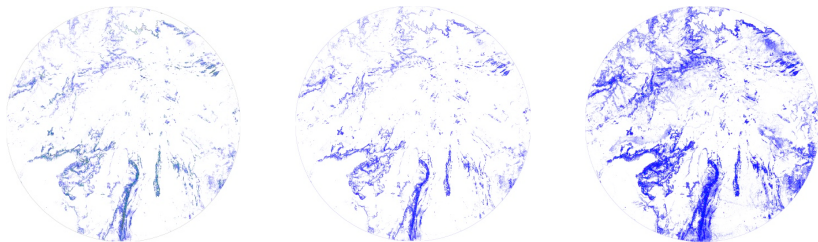


Resultados Experimentales

Caracterización de la Plataforma de Ejecución

Nombre	ELEANORRIGBY
Procesador	Intel®Xeon CPU X5680
Memoria	96GB
Clock	3.33GHz
Número de Procesadores	2
Cores por Procesador	6
GPU	NVIDIA TESLA C2070
Número de GPUs	2
CUDA cores por GPU	448
GPU Memory por GPU	6GB GRR3
Versión CUDA	4.0

Mapas hídricos

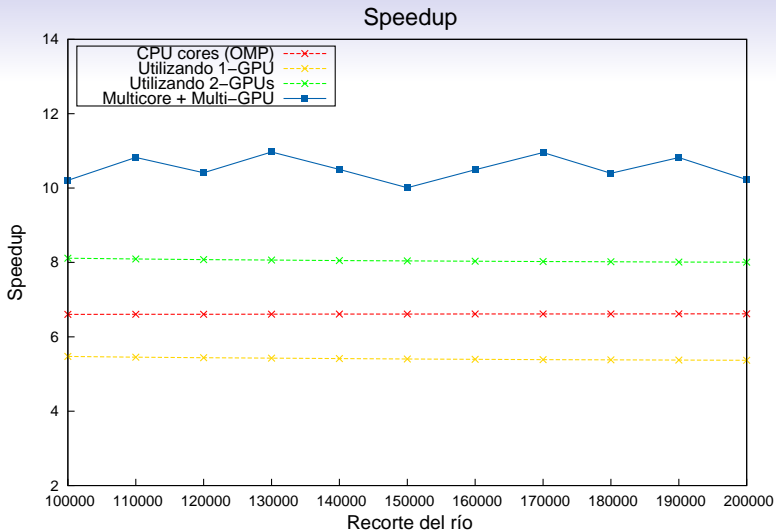


Mapas hídricos del Rio São Francisco para escorrentía con 55%, 60% y 90% de resolución.

Modelado Hídrico de Ríos

Tabla con los tiempos de ejecución (en segundos) variando el tamaño del problema (recorte del río).

Recorte del río	Sec.	OMP	1-GPU	2-GPUs	híbrido
20 mil	29.07	14.46	4.94	12.67	10.13
40 mil	113.15	47.24	25.53	14.56	12.00
60 mil	252.20	59.27	58.31	33.81	29.52
80 mil	446.20	87.66	107.10	56.42	52.69
100 mil	695.16	105.29	169.03	75.38	66.50
120 mil	999.08	151.21	245.05	95.71	85.96
140 mil	1357.96	205.42	335.17	130.40	118.07
160 mil	1771.81	267.91	439.28	170.44	141.83
180 mil	2240.61	338.69	557.69	215.85	167.23
200 mil	2764.37	417.75	690.10	266.61	194.29



Valores de *speedup* conseguido para cada configuración algorítmica respecto del tamaño del problema (recorte del río).

● Conclusiones del Modelado Hídrico de Ríos

- El modelado hídrico de ríos nos ha permitido establecer criterios para **comparar cuencas** de distintas dimensiones.
- Se ha aportado al problema el modelado computacional paralelo que permite la **reducción del tiempo de ejecución** requerido por una simulación con características hidrológicas.

Contenido

Introducción y Contextualización de la investigación

Planteamiento, objetivos, ámbito del trabajo, problemas a tratar, estado del arte y herramientas computacionales

Problemas Aplicados a Ingeniería Agrícola

Representación del Relieve

Polinomios Matriciales

Variables Meteorológicas del Clima

Modelado Hídrico de Ríos

Autooptimización en Aplicaciones Paralelas

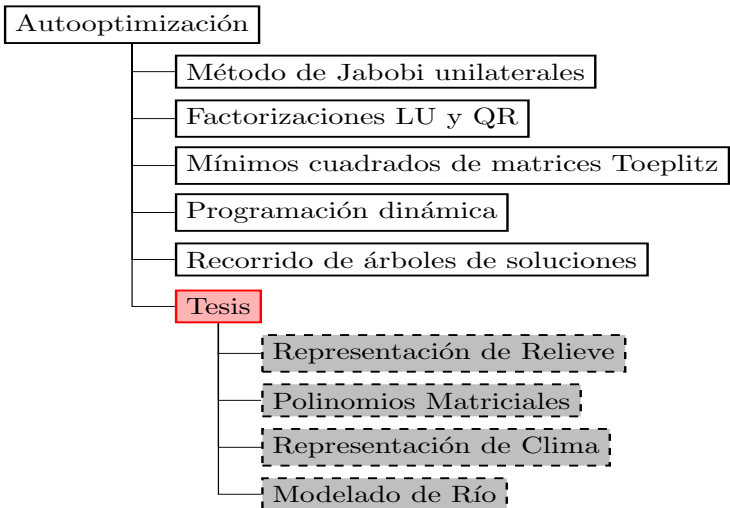
Conclusiones Generales y Aportaciones

Autooptimización en Aplicaciones Paralelas

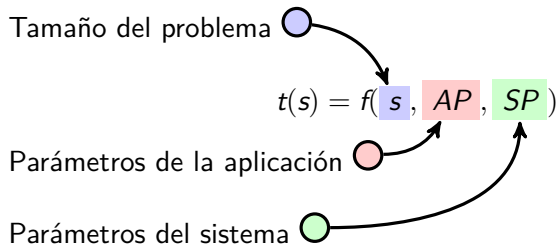
Autooptimización en Aplicaciones Paralelas

Se aplican técnicas que permiten a las rutinas diseñadas **adaptarse automáticamente** a las características del sistema paralelo donde van a ser instaladas y ejecutadas con el objeto de obtener la versión más cercana posible a la óptima a un bajo coste.

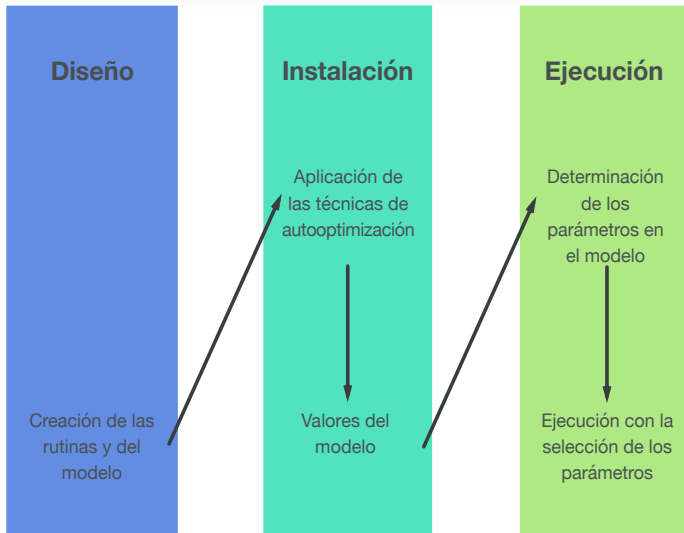
Experiencia en la aplicación de técnicas



Parametrización del modelo del tiempo de ejecución



Metodología Autooptimización en Aplicaciones Paralelas



Aplicación de la metodología

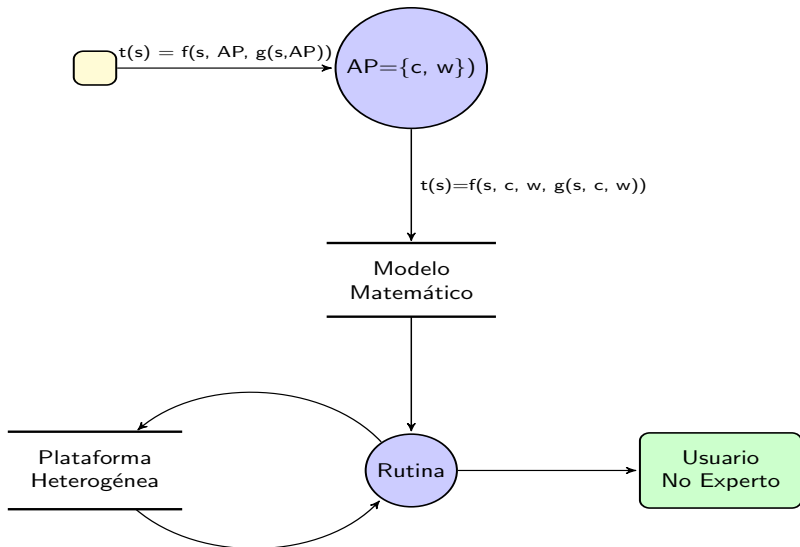
Así pues, en nuestros problemas, el conjunto de **parámetros algorítmicos** son

$$AP = \{c, w\} ,$$

luego, el **tiempo total de ejecución** es

$$t(s) = f\left(s, AP, SP \right) = f\left(s, c, w, g(s, c, w) \right) .$$

Ciclo de vida de la autooptimización



Aplicación de la metodología

Modelo Empírico


Se ha utilizado el **modelo teórico del tiempo de ejecución** para obtener la división óptima del problema en el sistema heterogéneo.

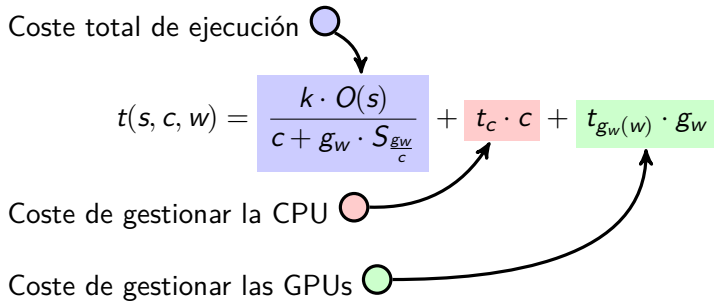
Modelo Experimental


La técnica desarrollada consiste en **lanzar un conjunto de ejecuciones con la rutina híbrida** que representa a la propia aplicación sobre un conjunto de recursos. Las ejecuciones toman valores de dicho conjunto y varían, a su vez, las porciones de reparto de trabajo que se asignan al subsistema CPU y GPU.

Modelo Empírico

$$t_{paralelo} = t_{rutina} + t_{CPU} + t_{GPU}$$

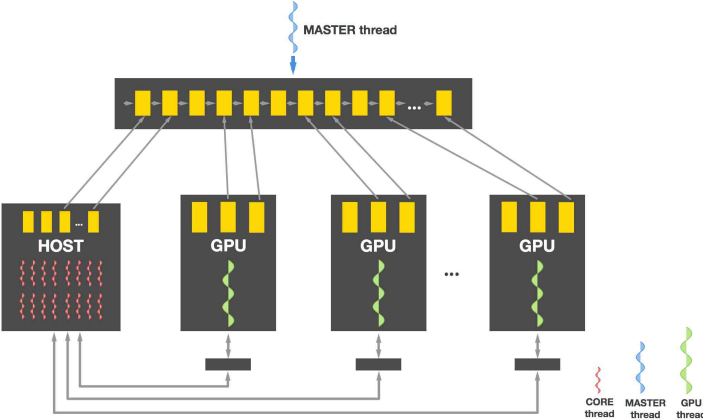
Coste total de ejecución 

$$t(s, c, w) = \frac{k \cdot O(s)}{c + g_w \cdot S \frac{g_w}{c}} + t_c \cdot c + t_{g_w(w)} \cdot g_w$$


Coste de gestionar la CPU 

Coste de gestionar las GPUs 

Modelo Experimental



Elección de los parámetros *AP* para 3 tipos de usuarios

Tratando a los **usuarios** de hasta 3 maneras diferentes podemos adecuar nuestras herramientas de autooptimización y predecir la optimización que cabe esperar según cada uno de estos 3 tipos de usuario.

UsuarioAT: **Utiliza** la autooptimización y **aplica** los parámetros óptimos obtenidos en el proceso de instalación con el modelo teórico presentado anteriormente.

UsuarioEO: **Utiliza** la autooptimización y **obtiene** el tiempo de ejecución **sustituyendo** los parámetros algorítmicos para el entorno de ejecución elegido.

UsuarioNE: No **utiliza** la autooptimización y está compuesto por usuarios no expertos en paralelismo. Este tipo de usuario **utiliza** 2-GPUs eligiendo de forma aleatoria los parámetros *AP*.

Modelo Empírico x Experimental

- **ELEANORRIGBY:**

- Número de hilos (c)={24}
- Workload (w): (GPU, GPU, CPU) = {(40%, 40%, 20%)}

- **MICROMACHIN:**

- Número de hilos (c)={16}
- Workload (w): (GPU, GPU, GPU, GPU, CPU) = {(20%, 20%, 20%, 20%, 20%)}

- **ELEANORRIGBY:**

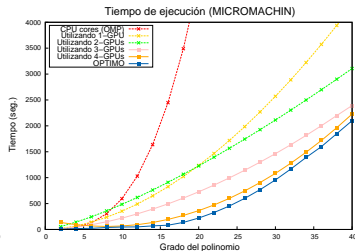
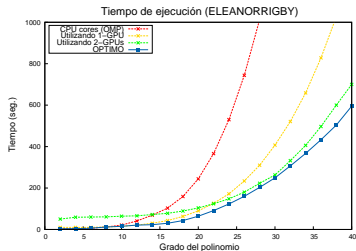
- Número de hilos (c)={24}
- Workload (w): (GPU, GPU, CPU) = {(45%, 45%, 10%)}

- **MICROMACHIN:**

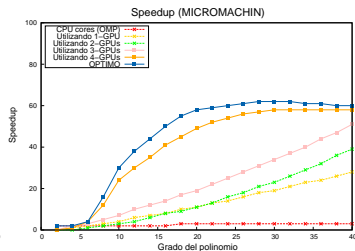
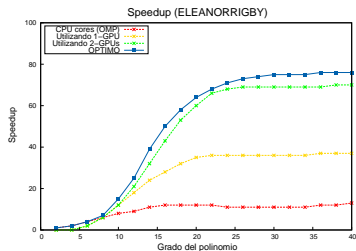
- Número de hilos (c)={16}
- Workload (w): (GPU, GPU, GPU, GPU, CPU) = {(22%, 22%, 22%, 22%, 12%)}

AUTO: Representación del Relieve

Tiempo de Ejecución

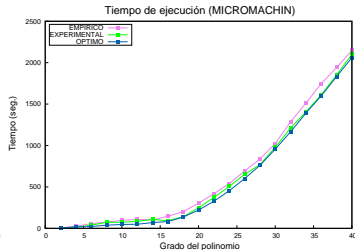
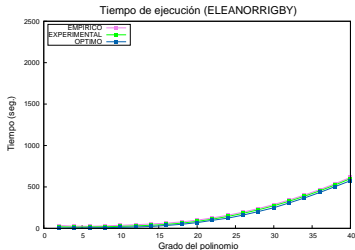


Speedup

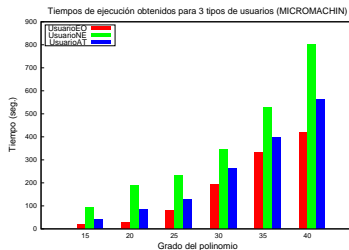
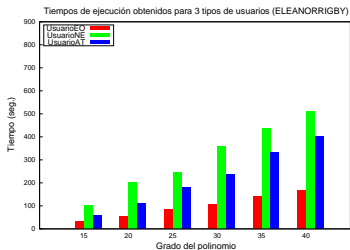


AUTO: Representación del Relieve

Discrepancia

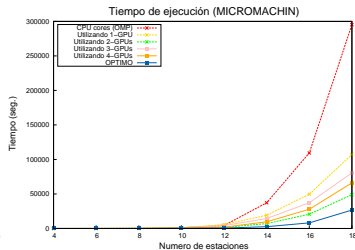
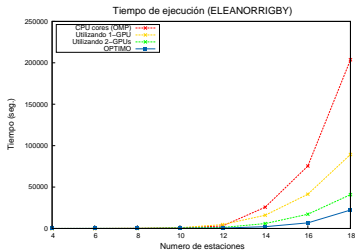


Tipos de usuarios

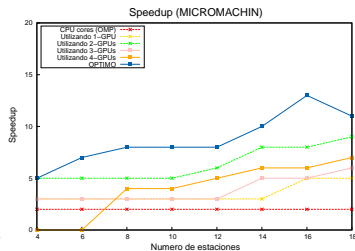
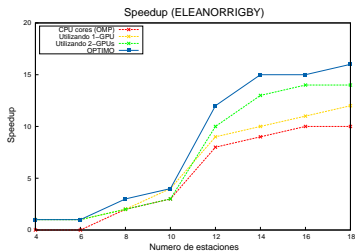


AUTO: Representación del Clima

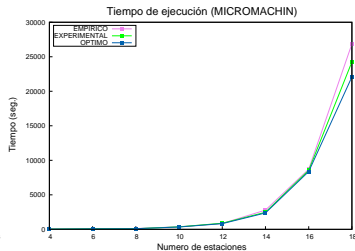
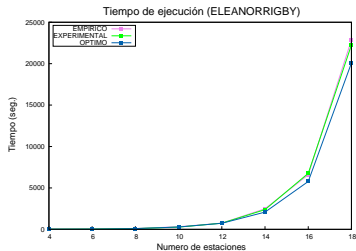
Tiempo de Ejecución



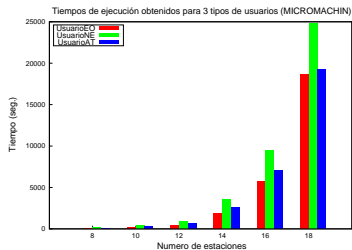
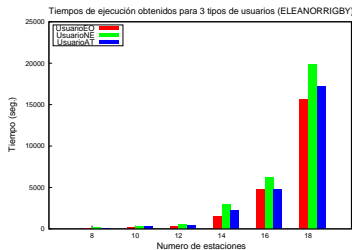
Speedup



Discrepancia

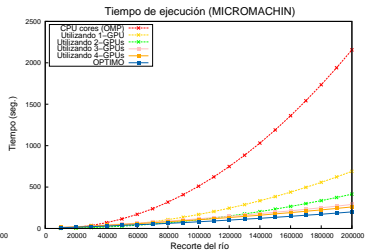
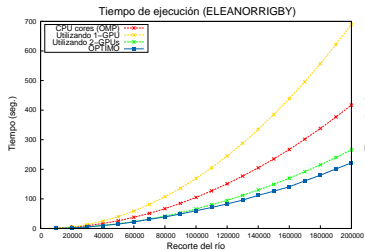


Tipos de usuarios

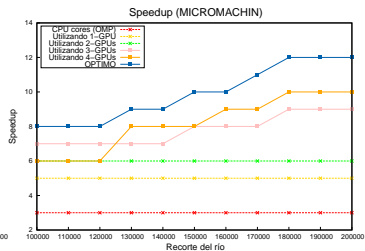
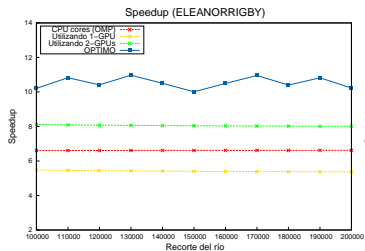


AUTO: Modelado Hídrico de Ríos

Tiempo de Ejecución

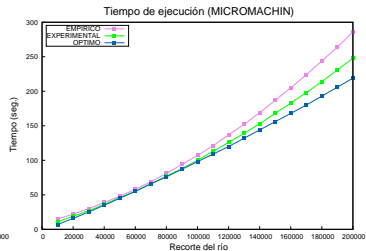
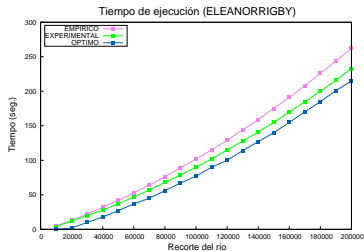


Speedup

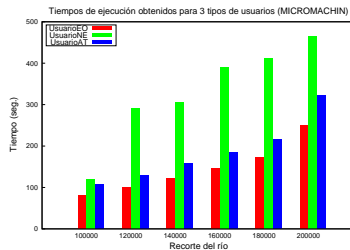
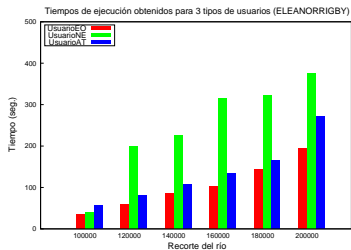


AUTO: Modelado Hídrico de Ríos

Discrepancia



Tipos de usuarios



- **Conclusiones de Autooptimización en Aplicaciones Paralelas**

- Los **mejores valores** obtenidos se almacenan para ser utilizados en tiempo de ejecución y poder así determinar la forma de repartir las tareas entre CPU y GPU.
- El modelo contiene parámetros que reflejan el **comportamiento** de la rutina en el sistema, y los valores de estos parámetros se estiman para cada sistema concreto por medio de experimentos.
- Podemos afirmar que, en general, con los dos modelos se obtienen distribuciones muy cercanas a la **óptima** en los dos sistemas heterogéneos en los que se ha experimentado.

- **Conclusiones de Autooptimización en Aplicaciones Paralelas**

- Los **mejores valores** obtenidos se almacenan para ser utilizados en tiempo de ejecución y poder así determinar la forma de repartir las tareas entre CPU y GPU.
- El modelo contiene parámetros que reflejan el **comportamiento** de la rutina en el sistema, y los valores de estos parámetros se estiman para cada sistema concreto por medio de experimentos.
- Podemos afirmar que, en general, con los dos modelos se obtienen distribuciones muy cercanas a la **óptima** en los dos sistemas heterogéneos en los que se ha experimentado.

- **Conclusiones de Autooptimización en Aplicaciones Paralelas**

- Los **mejores valores** obtenidos se almacenan para ser utilizados en tiempo de ejecución y poder así determinar la forma de repartir las tareas entre CPU y GPU.
- El modelo contiene parámetros que reflejan el **comportamiento** de la rutina en el sistema, y los valores de estos parámetros se estiman para cada sistema concreto por medio de experimentos.
- Podemos afirmar que, en general, con los dos modelos se obtienen distribuciones muy cercanas a la **óptima** en los dos sistemas heterogéneos en los que se ha experimentado.

Contenido

Introducción y Contextualización de la investigación

Planteamiento, objetivos, ámbito del trabajo, problemas a tratar, estado del arte y herramientas computacionales

Problemas Aplicados a Ingeniería Agrícola

Representación del Relieve

Polinomios Matriciales

Variables Meteorológicas del Clima

Modelado Hídrico de Ríos

Autooptimización en Aplicaciones Paralelas

Conclusiones Generales y Aportaciones

Conclusiones Generales y Aportaciones

Conclusiones y Aportaciones

- En una **primera** aportación hemos realizado una implementación del método de regresión polinomial utilizado para resolver el problema de la **Representación del Relieve** para GPUs.
- La **segunda** aportación ha consistido en el desarrollo de algoritmos paralelos, tanto para CPUs como para GPUs, para la **Representación de Variables Meteorológicas del Clima**.

Conclusiones y Aportaciones

- Como **tercera** aportación hemos acelerado el proceso de construcción de la matriz del sistema que describe el **Modelo Hídrico de Ríos** mediante la utilización de cores CPU y GPUs.
- Consideramos la **última** de las aportaciones de la tesis como una de las más significativas debido al impacto real que puede tener en el futuro aprovechamiento de este trabajo. Con ello logramos una **Adaptación Automática** del software a la arquitectura destino que los usuarios finales puedan utilizar de manera sencilla y eficiente.

Otros problemas

Modelado matemático de control biológico de plagas de caña de azúcar, desarrollo de un modelo lagrangiano de transporte de aceite, etc.

Explotar en totalidad la heterogeneidad del sistema

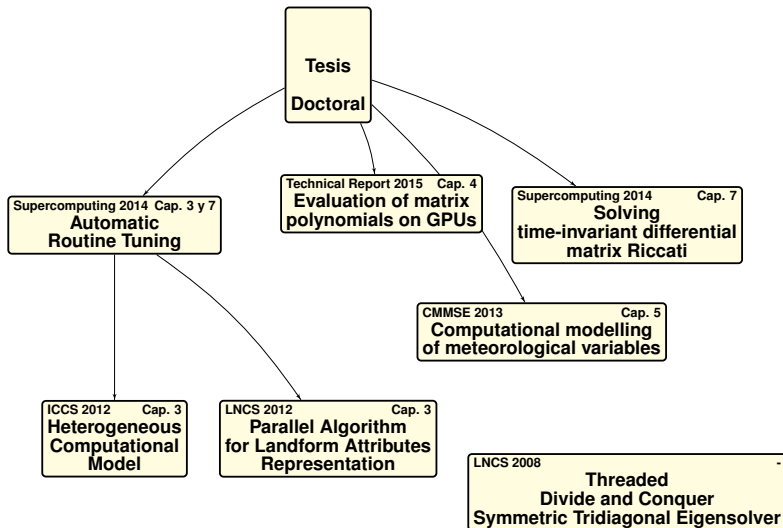
Nuestra implementación está diseñada para la división y asignación de tareas a **diferentes recursos computacionales** en multicores con varias GPUs.

Nuevos modelos de autooptimización

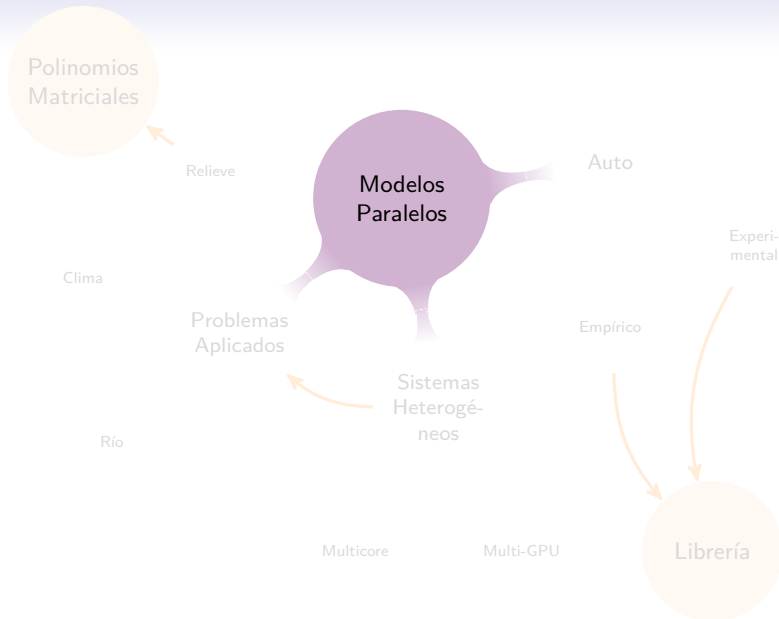
Se ha comprobado en el anterior que los modelos obtenidos puede ser mejorables en el sentido de **reducir el espacio de búsqueda** en el proceso de experimentación.

Extensión a nuevas configuraciones hardware

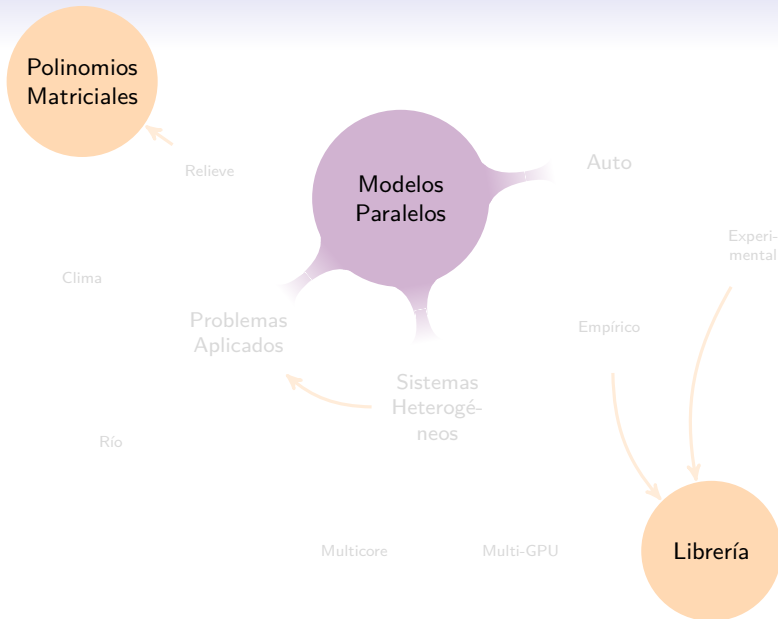
Otros **aceleradores hardware** como el Intel Many-Integrated-Core Xeon Phi, u otros tipos de GPUs no de NVIDIA.



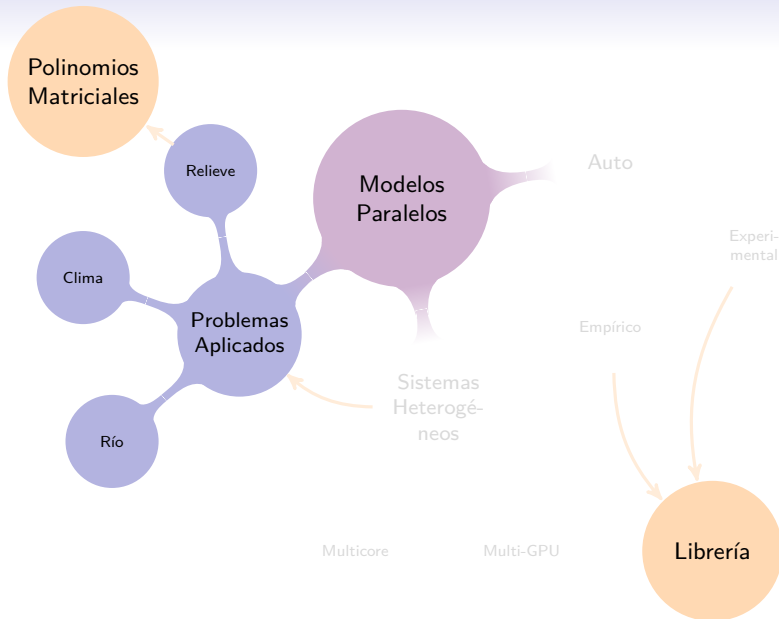
Resumen de la tesis



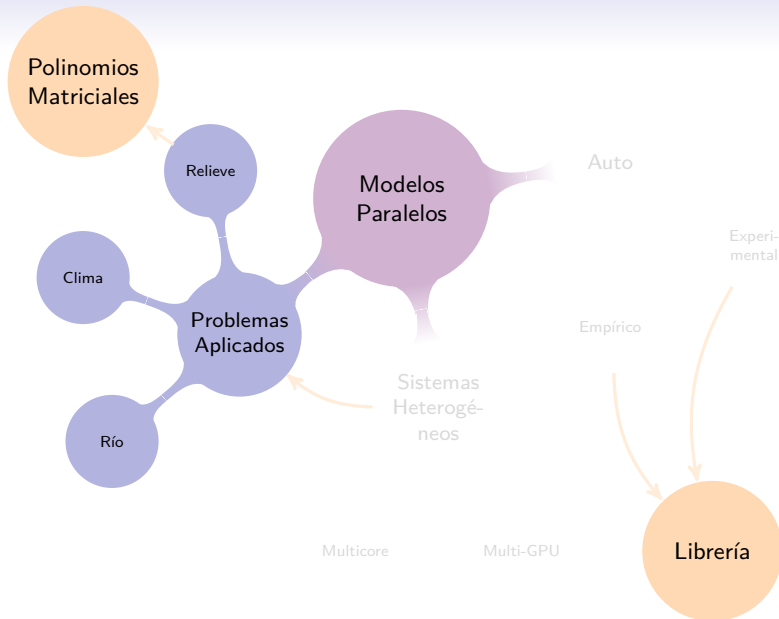
Resumen de la tesis



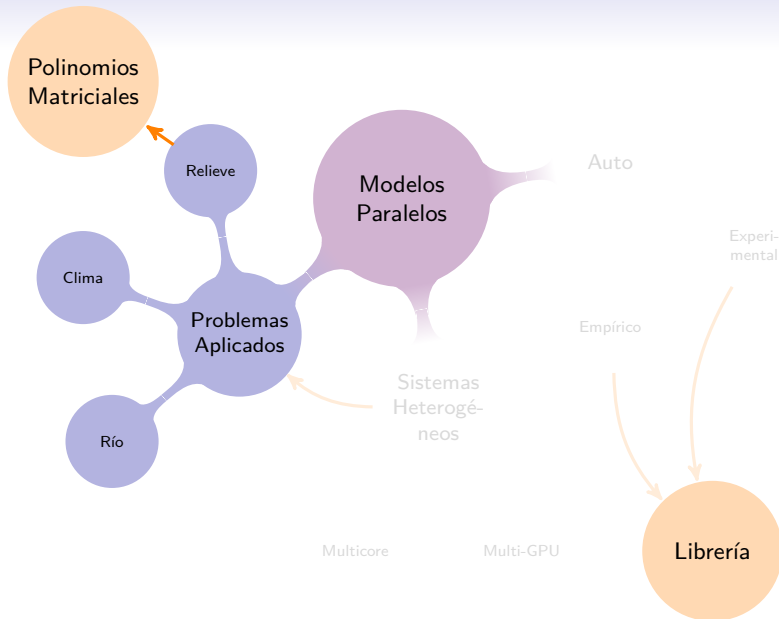
Resumen de la tesis



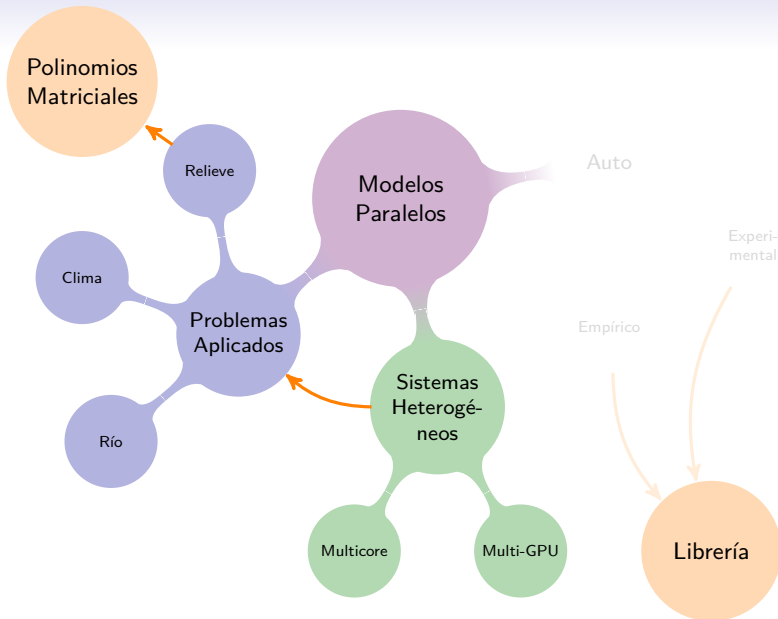
Resumen de la tesis



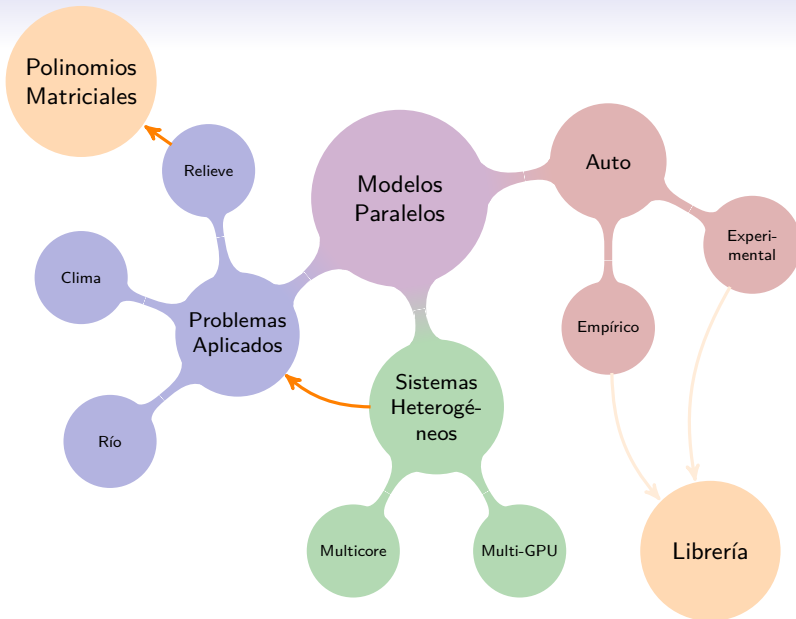
Resumen de la tesis



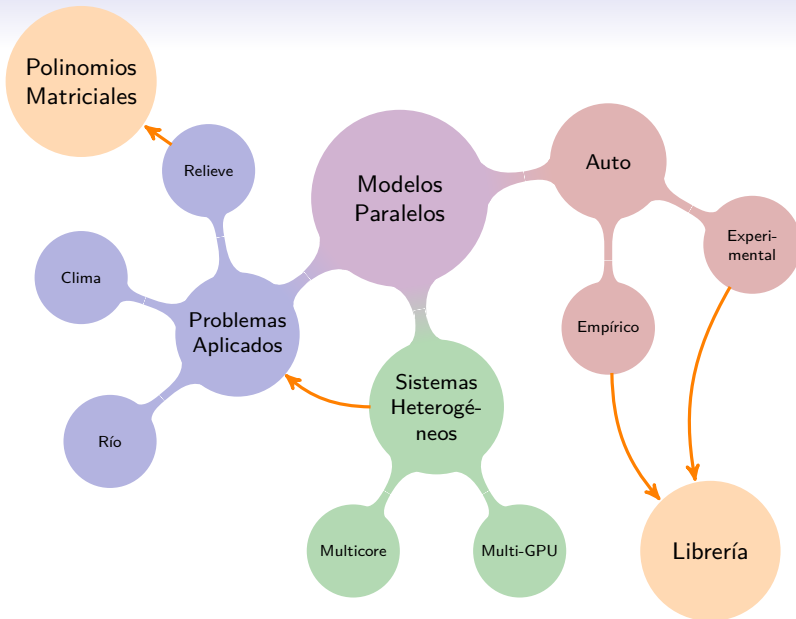
Resumen de la tesis



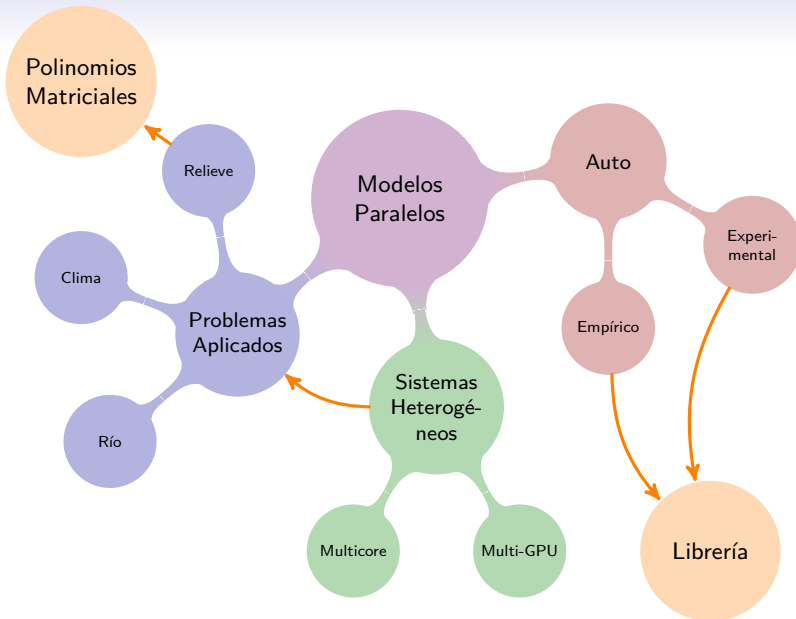
Resumen de la tesis



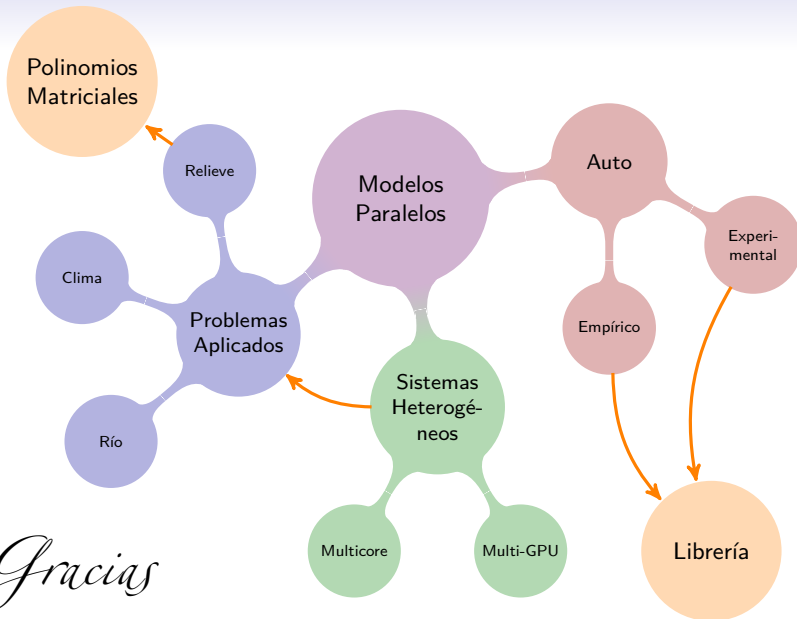
Resumen de la tesis



Resumen de la tesis



Resumen de la tesis



Gracias