Desarrollo, evaluación y autooptimización de rutinas paralelas híbridas de cálculo de funciones de Green

Proyecto Fin de Carrera

Tomás Ramírez García

Universidad de Murcia
Facultad de Informática
Ingeniería en Informática

Murcia, Septiembre de 2012

Contenidos

Contenidos

Punto de partida

Conjunto de rutinas paralelas de cálculo de funciones de Green en problemas de electromagnetismo:

- Unidimensionales:
  - OpenMP (grano fino)
  - CUDA (grano fino)
  - OpenMP (grano grueso)
  - OpenMP+CUDA (grano grueso + grano fino)
- Bidimensionales:
  - OpenMP (grano fino)
  - MPI (grano fino)
  - CUDA (grano fino)

Objetivos y metodología

Estudiar la aceleración de una aplicación científica basada en un problema real aplicando diferentes herramientas de programación paralela:

- Evaluar rutinas anteriores en otros sistemas
- Combinar MPI, OpenMP y CUDA
- Diseñar un mecanismo de autooptimización

Contenidos

Herramientas software

OpenMP (*Open Multi-Processing*)

Un estándar para programación en memoria compartida

MPI (*Message Passing Interface*)

Un estándar de paso de mensajes

CUDA (*Compute Unified Device Architecture*)

Computación de propósito general sobre GPUs

Herramientas hardware

Servidor *luna* (PCGUM)

4 cores memoria compartida + 1 GPU $\times$ 112 CUDA cores

Estación de trabajo *geatpc2* (GEAT UPCT)

8 cores memoria compartida + 1 GPU $\times$ 96 CUDA cores

Cluster *hipatia* (SEDIC-SAIT UPCT)

18 nodos memoria distribuida (152 cores), utilizados:

- 1 nodo $\times$ 16 cores (menos potente)
- 2 nodos $\times$ 8 cores (más potentes)

Servidores *marte* y *mercurio* (PCGUM)

2 nodos memoria distribuida $\times$
(6 cores memoria compartida + 2 GPUs $\times$ 512 CUDA cores)

Contenidos

Funciones de Green unidimensionales

n: Número de puntos fuente

m: Número de puntos observación

nmod: Número de modos (espectral y espacial)

```
1  GF_Capa2Dt_Tex_conv(n, m, nmod, ...) {
2    para cada punto fuente de 1 hasta n { // Grano grueso (OpenMP)
3      para cada punto observacion de 1 hasta m {
4        repetir nmod iteraciones {  // Grano fino (OpenMP/CUDA)
5          Operaciones exponenciales;
6          Operaciones trignometricas;
7        }
8        Metodo Aceleracion Kummer;
9      }
10   }
11 }
```

Funciones de Green unidimensionales

Secuencial

$$t(n, m, nmod) = nm\,(nmod\,(E + T) + K)$$

OpenMP (grano fino)

$$t(n, m, nmod, h) = nm\left(\frac{nmod\,(E + T)}{h} + K\right)$$

CUDA (grano fino)

$$t(n, m, nmod) = nm\,(nmod\,(E_{GPU} + T_{GPU}) + K)$$

Funciones de Green unidimensionales

OpenMP (grano grueso)

$$t(n, m, nmod, h) = \frac{nm\,(nmod\,(E + T) + K)}{h}$$

OpenMP+CUDA (grano grueso + grano fino)

$$t(n, m, nmod, h) = \frac{nm\,(nmod\,(E + T) + K)}{h + S_{GPU/CPU}}$$

Funciones de Green bidimensionales

x: Número de puntos eje *X* (observación)
y: Número de puntos eje *Y* (observación)
nmod: Número de modos (parte espectral)
n: Número de imágenes eje *X* (parte espacial)
m: Número de imágenes eje *Y* (parte espacial)

```
1  GF_wg_ewald_xy_ord(x, y, nmod, n, m, ...) {
2    para cada punto en plano observacion de 1 hasta x {
3      para cada punto en plano observacion  de 1 hasta y {
4        repetir nmod iteraciones {  // Grano fino (OpenMP)
5          Parte espectral de la funcion de Green;
6        }
7
8        para cada imagen en el eje Y de -m hasta m {  // Grano fino (OpenMP/CUDA/MPI)
9          para cada imagen en el eje X de -n hasta n {
10           Parte espacial de la funcion de Green;
11         }
12       }
13     }
14   }
15 }
```

Funciones de Green bidimensionales

Secuencial

$$t(x, y, nmod, n, m) = xy\,(nmod \cdot S_1 + (2m+1)(2n+1)S_2)$$

OpenMP (grano fino)

$$t(x, y, nmod, n, m, h) = xy\left(\frac{nmod \cdot S_1 + (2m+1)(2n+1)S_2}{h} + R_{OMP}(h)\right)$$

CUDA (grano fino)

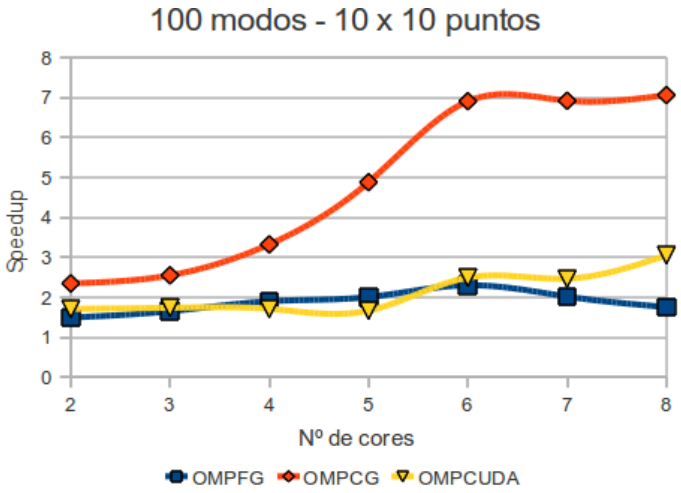
$$t(x, y, nmod, n, m) = xy\,(nmod \cdot S_1 + (2m+1)(2n+1)\,(S_{2(GPU)} + R))$$

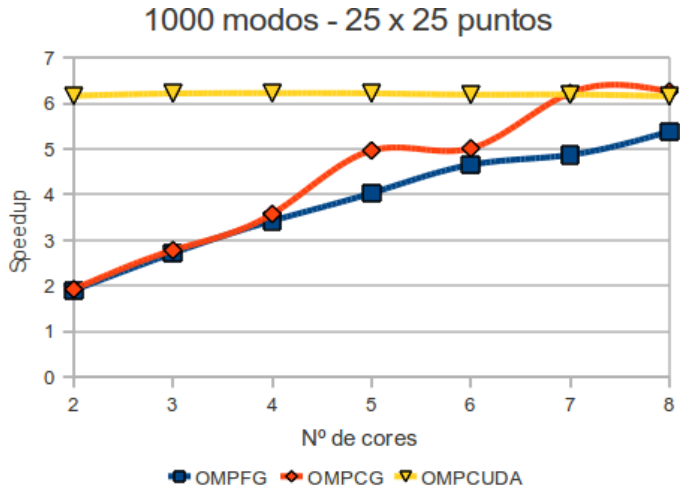
MPI (grano fino)

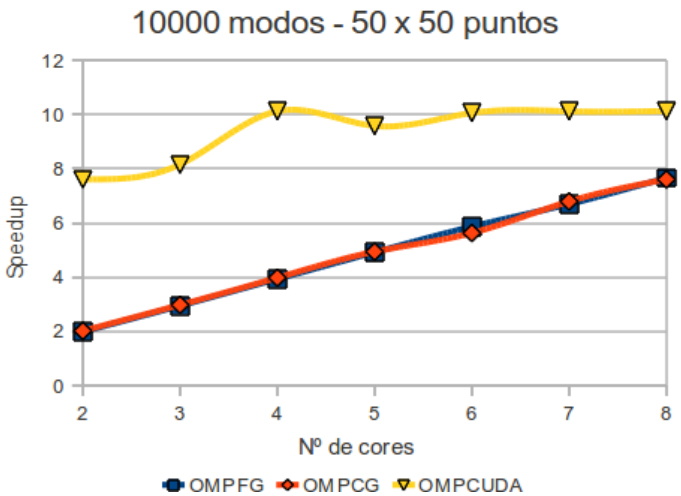
$$t(x, y, nmod, n, m, p) = xy\left(nmod \cdot S_1 + \frac{(2m+1)(2n+1)S_2}{p} + R_{MPI}(p)\right)$$

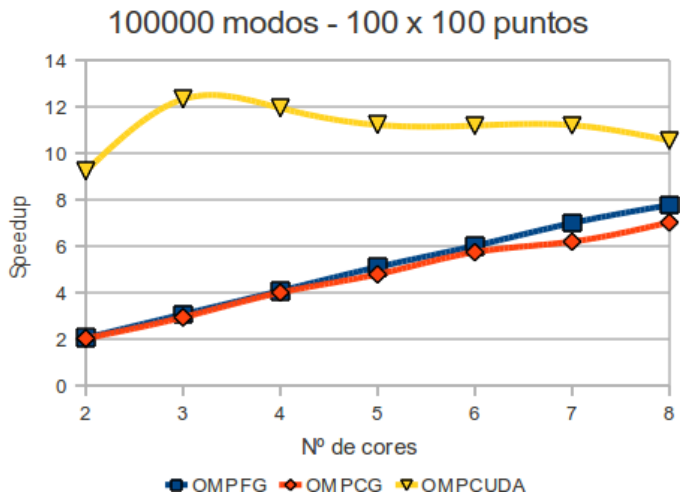
Contenidos

1D: Evolución del speedup (*geatpc2*)









1D: Rutinas preferidas

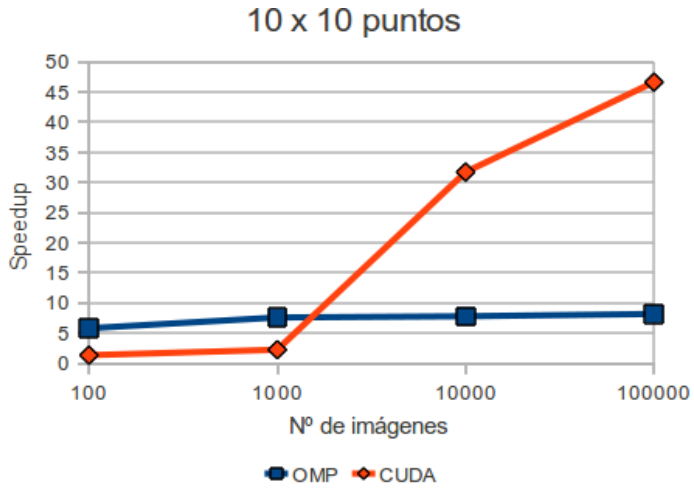
luna

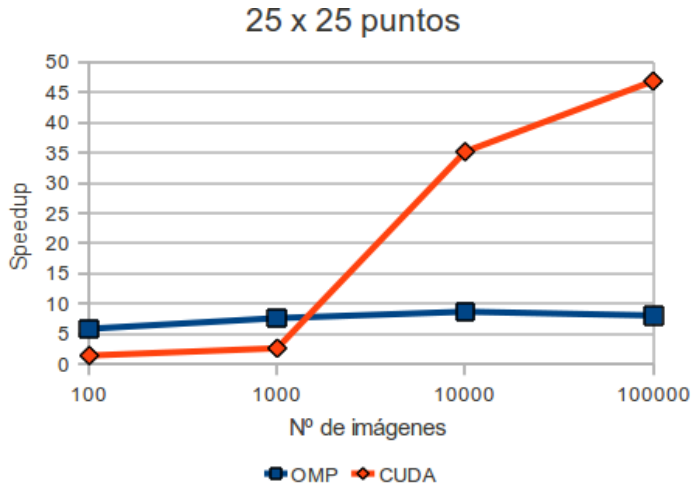
$n \times m \setminus nmod$	100	1000	10000	100000
10×10	OMPCG/4	OMPCG/4	OMPFG/4	OMPCUDA/4
25×25	OMPCG/4	OMP[F,C]G/4	OMPCUDA/4	OMPCUDA/4
50×50	OMPCG/4	OMP[F,C]G/4	OMPCUDA/4	OMPCUDA/4
100×100	OMPCG/4	OMPCUDA/4	OMPCUDA/4	OMPCUDA/4

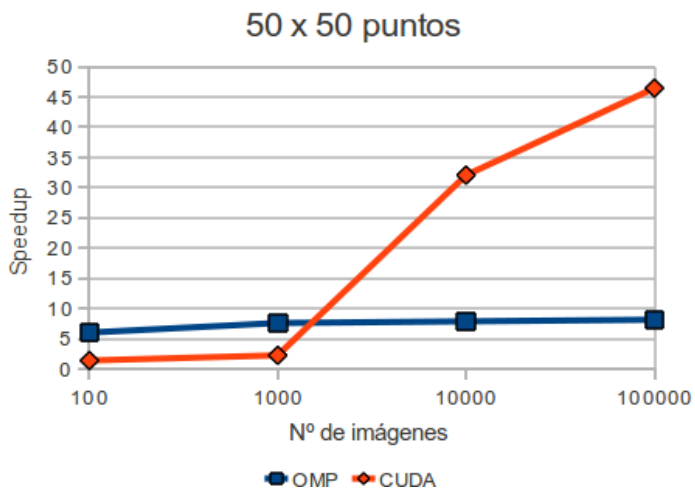
geatpc2

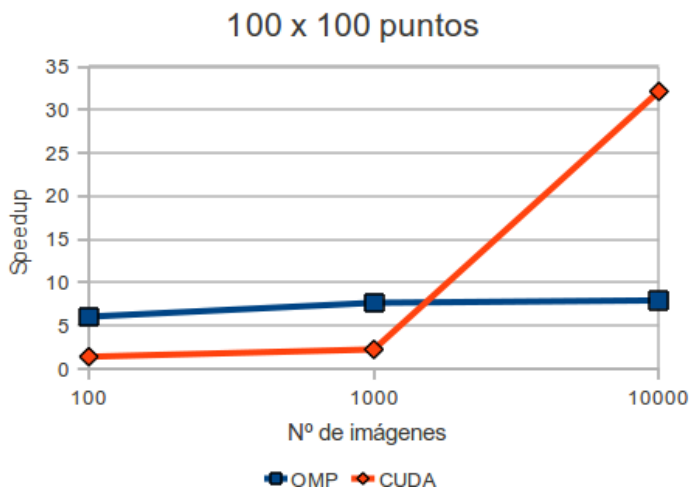
$n \times m \setminus nmod$	100	1000	10000	100000
10×10	OMPCG/8	OMPFG/8	OMPCUDA/6	OMPCUDA/5
25×25	OMPCG/8	OMPCG/8	OMPCUDA/5	OMPCUDA/6
50×50	OMPCUDA/5	OMPCUDA/3	OMPCUDA/8	OMPCUDA/6
100×100	OMPCG/8	OMPCUDA/4	OMPCUDA/8	OMPCUDA/3

2D: Sistema completo (*geatpc2*)

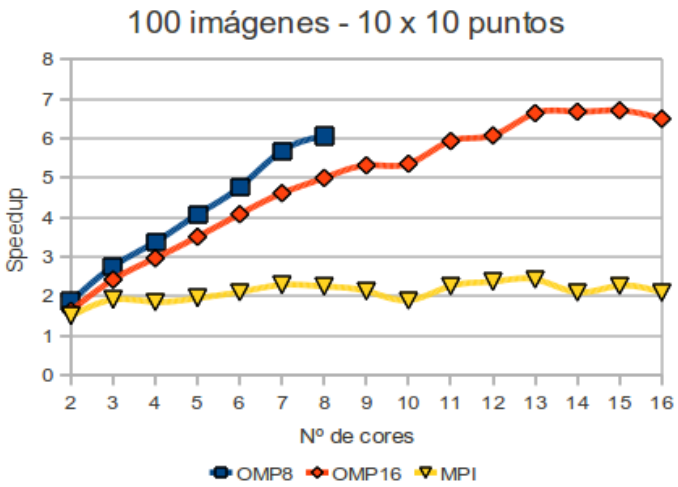


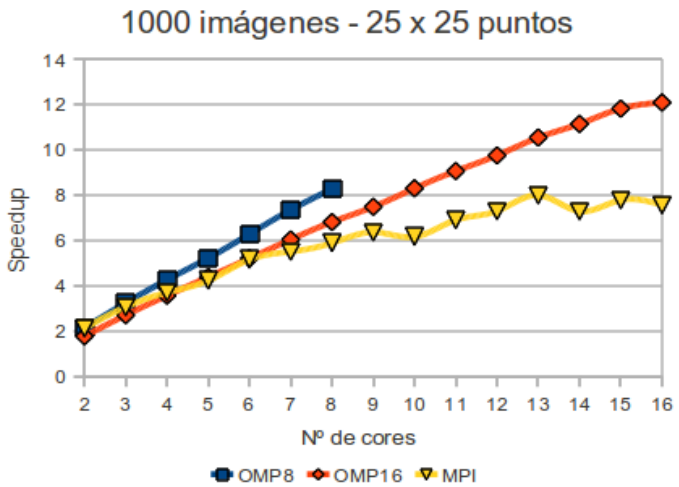


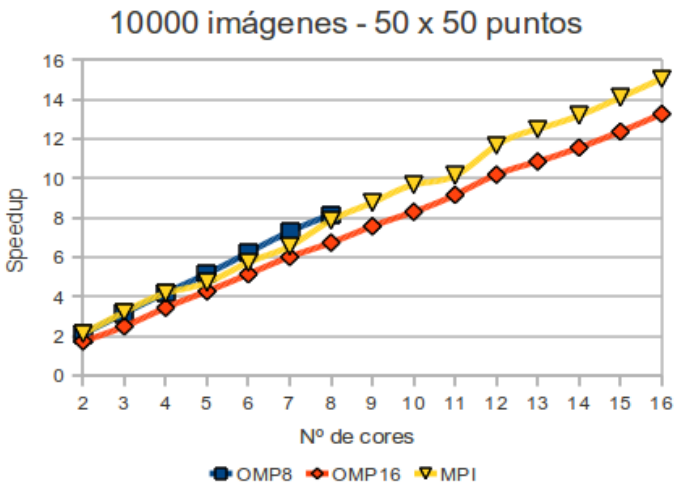


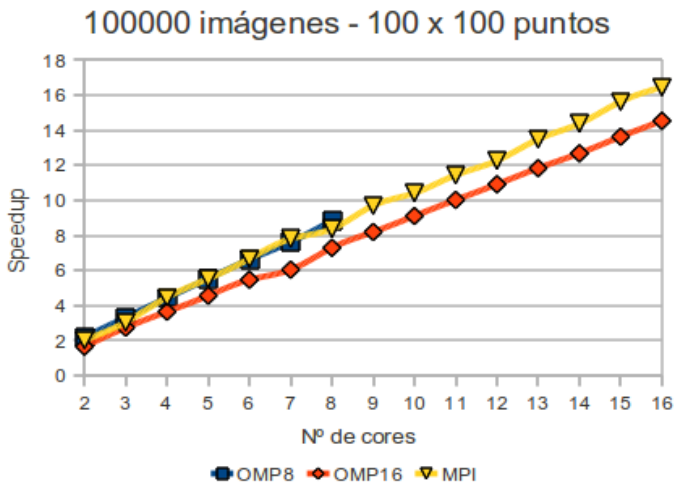


2D: Evolución del speedup (*hipatia*)









2D: Rutinas preferidas

luna

$x \times y \setminus nm$	100	1000	10000	100000
10×10	OMP/4	CUDA	CUDA	CUDA
25×25	OMP/4	CUDA	CUDA	CUDA
50×50	OMP/4	CUDA	CUDA	CUDA
100×100	OMP/4	CUDA	CUDA	

geatpc2

$x \times y \setminus nm$	100	1000	10000	100000
10×10	OMP/8	OMP/8	CUDA	CUDA
25×25	OMP/8	OMP/8	CUDA	CUDA
50×50	OMP/8	OMP/8	CUDA	CUDA
100×100	OMP/8	OMP/8	CUDA	

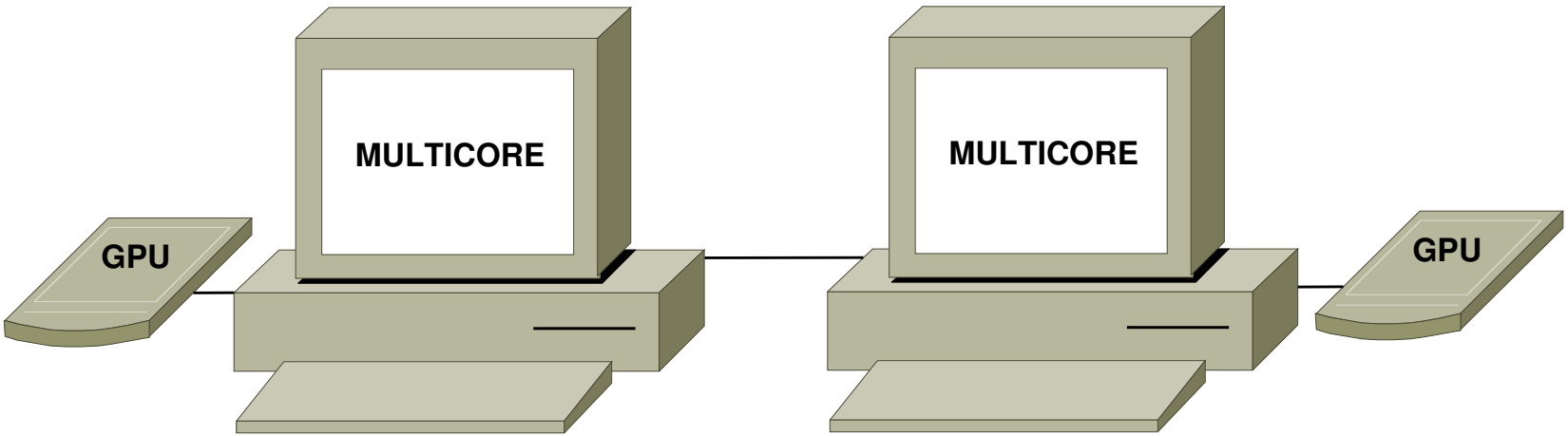
2D: Rutinas preferidas

hipatia

$x \times y \setminus nm$	100	1000	10000	100000
10×10	OMP16/15	OMP16/16	MPI/16	MPI/15
25×25	OMP16/15	OMP16/16	MPI/16	MPI/16
50×50	OMP16/14	OMP16/16	MPI/16	MPI/16
100×100	OMP16/14	OMP16/16	MPI/16	MPI/16

Contenidos

Sistema computacional jerárquico



- MPI+OpenMP+CUDA
- Funciones de Green bidimensionales: alto coste

Diseño e implementación

```
1  GF_wg_ewald_xy_ord(x, y, nmod, n, m, ...) {
2    para cada punto en plano observacion de 1 hasta x { // Grano grueso (MPI+OpenMP)
3      para cada punto en plano observacion  de 1 hasta y {
4        repetir nmod iteraciones {
5          Parte espectral de la funcion de Green;
6        }
7
8        para cada imagen en el eje Y de -m hasta m {  // Grano fino (CUDA)
9          para cada imagen en el eje X de -n hasta n {
10           Parte espacial de la funcion de Green;
11         }
12       }
13     }
14   }
15 }
```

Diseño e implementación

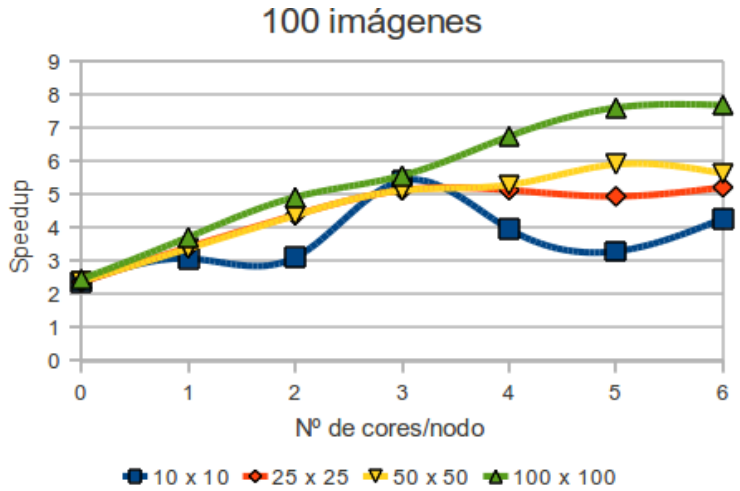
Modelo teórico

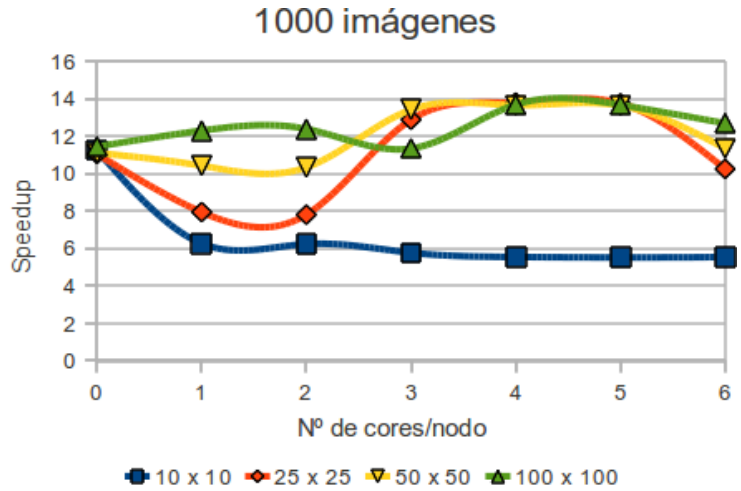
$$t(x, y, nmod, n, m, p, h, g) = \frac{xy(nmod \cdot S_1 + (2m+1)(2n+1)S_2)}{p(h + gS_{GPU/CPU})} + G_{MPI}(x, y, p)$$

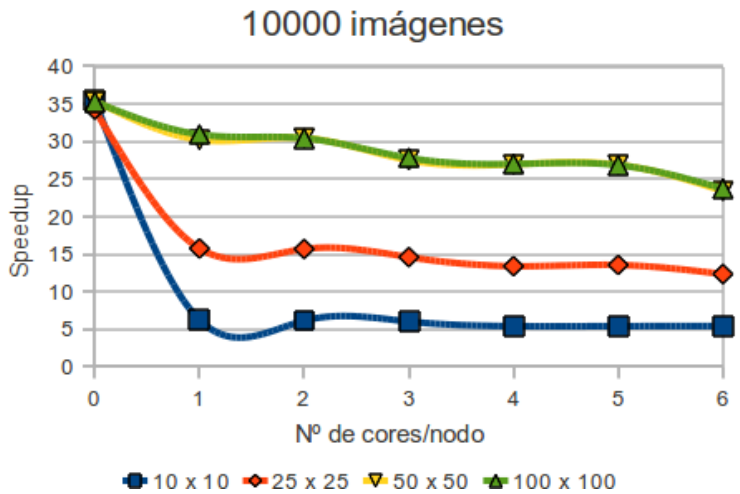
Una rutina $\rightarrow$ múltiples versiones

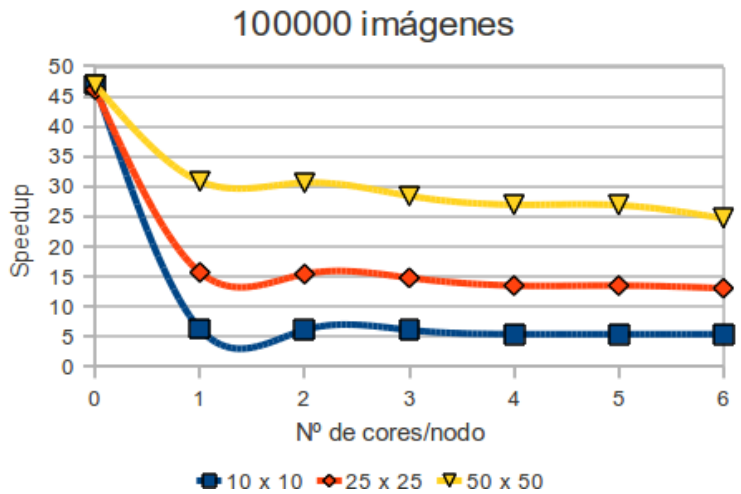
$p \setminus h+g$	$1+0$	$h+0$	$0+g$	$h+g$
1	SEQ	OMP	CUDA	OMP+CUDA
p	MPI	MPI+OMP	MPI+CUDA	MPI+OMP+CUDA

MPI+OpenMP+CUDA (1 kernel/nodo)

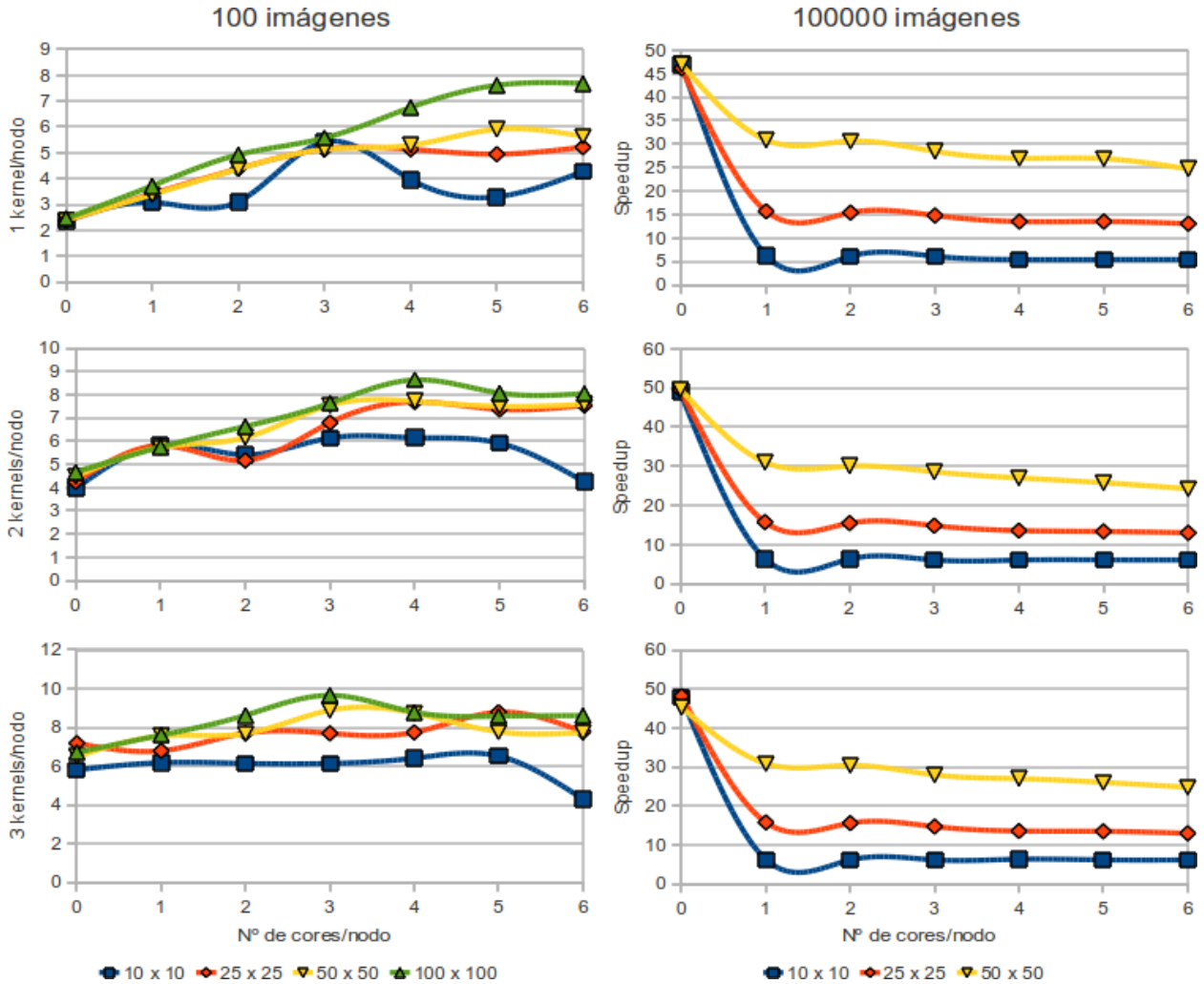








MPI+OpenMP+CUDA (1, 2 y 3 kernels/nodo)



Resumen de resultados

Configuraciones preferidas (*marte/mercurio*)

$x \times y \setminus nm$	100	1000	10000	100000
10×10	$2 \times (5+3)$	$2 \times (0+3)$	$2 \times (0+2)$	$2 \times (0+2)$
25×25	$2 \times (5+3)$	$2 \times (0+3)$	$2 \times (0+3)$	$2 \times (0+2)$
50×50	$2 \times (3+3)$	$2 \times (0+3)$	$2 \times (0+3)$	$2 \times (0+2)$
100×100	$2 \times (3+3)$	$2 \times (3+3)$	$2 \times (0+2)$	$2 \times (0+2)$

Contenidos

Justificación

- Muchas rutinas: ¿cuál es más rápida?
- Depende de:
  - Parámetros de entrada
  - Características del sistema
  - Elementos de procesamiento (p, h y g)
- Elegir rutina óptima en tiempo de ejecución
- Funciones de Green bidimensionales: alto coste

Metodología general

Diseño

Implementación de la/s rutina/s y la parte de autotuning

Instalación

Obtención de información sobre el comportamiento de la/s rutina/s en un sistema concreto

Ejecución

Utilización de la/s rutina/s, eligiendo la mejor en base a información de instalación

Instalación

Conjunto de instalación

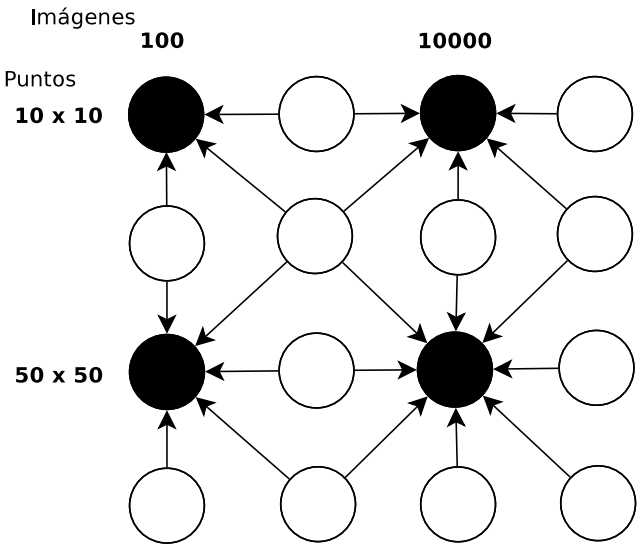
Diferentes combinaciones de $x \times y = \{10 \times 10, 50 \times 50\}$ y $nm = \{100, 10000\}$

Información de instalación

Rutina, configuración y speedup óptimos para conjunto de instalación

Elección implementación óptima

```
1  autotuning(x, y, n, m) {
2    speedupMax = 0;
3    para cada problema p en conjuntoInstalacion {
4      if ( p es vecino de problema(x, y, n, m) ) {
5        if (p.speedup > speedupMax) {
6          speedupMax = p.speedup;
7          preferido = p;
8        }
9      }
10   }
11
12   return (preferido.rutina, preferido.configuracion);
13 }
```



Contraste de resultados

hipatia

$x \times y$ *nm*	25×25 1000	25×25 100000	100×100 1000	100×100 100000
Autooptimización	0,245	9,614	4,477	160,766
Óptimo	0,151	9,614	2,116	160,766
Diferencia absoluta	0,094	0	2,361	0
Diferencia relativa	62,25 %	0 %	111,58 %	0 %

marte/mercurio

$x \times y$ *nm*	25×25 1000	25×25 100000	100×100 1000	100×100 100000
Autooptimización	0,155	5,012	1,706	87,814
Óptimo	0,114	5,012	1,656	79,453
Diferencia absoluta	0,041	0	0,050	8,361
Diferencia relativa	35,96 %	0 %	3,02 %	10,52 %

Contenidos

Vías futuras

- Funciones de Green tridimensionales
- Sistemas heterogéneos
- Utilizar las rutinas en aplicaciones de electromagnetismo
- Mecanismos de autooptimización. Remodelar rutinas
- Otros estándares: OpenCL
- Combinar con otros niveles de paralelismo: SSE

¿Preguntas?