

# Práctica 3

## Programación GPUs con directivas

Carlos García Sánchez

UCM

8 de mayo de 2020

- “Parallel Programming with OpenACC”, Rob Farber, 2016



# Outline

1 Objetivos

2 OpenACC

3 Ejemplos

4 Entrega



# Objetivos

- Familiarizarse con la programación por medio de directivas con OpenACC
- Evaluar las mejoras/speedup



# OpenACC

- Uso del compilador PGI con soporte OpenACC
  - Conocer las características de la GPU para poder compilar adecuadamente
- Preparados los **makefile**
- Entrega *Ejemplos 4,5 y 6*

## Terminal #1

```
carlos@7pico:~/GPU/Lab/Practica3/Ejemplo0$ make
gcc -O3 -lm -lrt -std=c99 hello.c -o hello.host.exe
pgcc -Minfo -fast -acc -ta=nvidia -tp=nehalem hello.c -o hello.pgi.exe

24, Loop not fused: function call before adjacent loop
Generated vector sse code for the loop
28, Generating copy(b[:])
Generating Tesla code
29, Loop is parallelizable
Accelerator kernel generated
29, #pragma acc loop gang, vector(128) /* blockIdx.x threadIdx.x */
32, Loop not vectorized/parallelized: potential early exits
```



# Ejemplo0

## ejemplo0.c

```
#ifdef _OPENACC
#include <openacc.h>
#endif
int main() {

#ifdef _OPENACC
    acc_init(acc_device_not_host);
    printf(" Compiling with OpenACC support \n");
#endif

    // Compute on the GPU if OpenACC support - host if not
    #pragma acc kernels copyout(b[0:N])
    for (int i = 0; i < N; i++) {
        b[i] = i;
    }
    ...
#ifdef _OPENACC
    acc_shutdown(acc_device_not_host);
#endif

    return 0;
}
```

## Terminal #1

```
carlos@tpicos:~/GPU/Lab/Practica3/Ejemplo0$ make pgi
pgcc -Minfo -fast -acc -ta=vidia -tp=nehalem hello.c -o hello.pgi.exe
24. Loop not fused: function call before adjacent loop
   Generated vector sse code for the loop
28. Generating copy(b[:])
   Generating Tesla code
29. Loop is parallelizable
   Accelerator kernel generated
   29. #pragma acc loop gang, vector(128) /* blockIdx.x threadIdx.x */
32. Loop not vectorized/parallelized: potential early exits
```



# Ejemplo1

axpy.c

```
...
#ifdef _OPENACC
#include <openacc.h>
#endif

int main (int argc, const char *argv[])
{
    ...
#ifdef _OPENACC
    acc_init(acc_device_not_host);
    int numdevices = acc_get_num_devices(acc_device_not_host);
    printf(" Compiling with OpenACC support NUM_DEVICES=%i\n",
        numdevices);
#endif

    ...
    // SAXPY
    t0 = get_time();
    #pragma acc ....
    #pragma acc ....
    for(i=0; i<n; i++)
        y_acc[i] = a*x_acc[i] + y_acc[i];
    ...

    return(1);
}
```



# Ejemplo2

## ■ Jacobi

- Método iterativo para resolución de ec. diferenciales
- Ej: solución para la ecuación de Laplace 2D ( $\nabla^2 f(x, y) = 0$ )

$$A_{k+1}(i, j) = \frac{A_k(i-1, j) + A_k(i+1, j) + A_k(i, j-1) + A_k(i, j+1)}{4}$$

### openacc\_jacobi.c

```
while ( error > tol && iter < iter_max ){
    error = 0.0;

    for( int j = 1; j < n-1; j++){
        for( int i = 1; i < m-1; i++ ){
            Anew[j][i] = 0.25 * ( A[j][i+1] + A[j][i-1]
                                + A[j-1][i] + A[j+1][i] );
            error = fmax( error, fabs(Anew[j][i] - A[j][i]));
        }
    }

    for( int j = 1; j < n-1; j++){
        for( int i = 1; i < m-1; i++ )
        {
            A[j][i] = Anew[j][i];
        }
    }

    if(iter % 100 == 0) printf("%5d, %0.6f\n", iter, error);

    iter++;
}
```



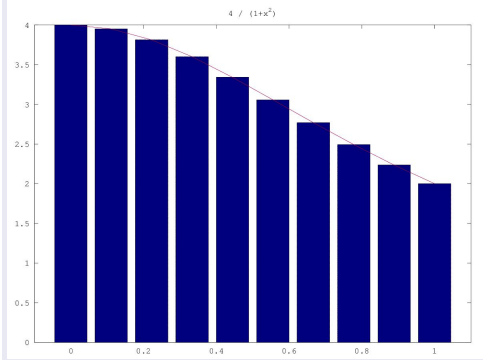
# Ejemplo3

## ■ Cálculo de PI

Suma de rectángulos

$$\pi = \int_0^1 \frac{4}{1+x^2} dx$$

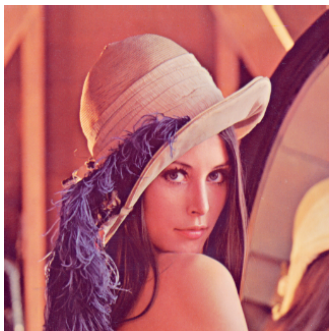
$$\pi \approx \sum_{i=0}^N F(x_i) \Delta x$$





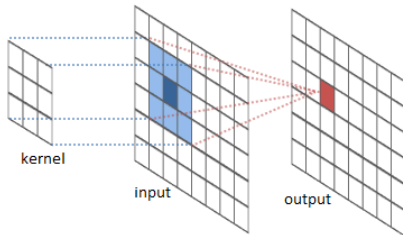
## Ejemplo4

- Tratamiento de imágenes: Detección de bordes
  - Basado en **convolución 2D**



## Ejemplo4

- Tratamiento de imágenes: Detección de bordes
  - Basado en **convolución 2D**



$$Im_{OUT}(i,j) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} Im_{IN}(i-1:i+1, j-1:j+1)$$



## Ejemplo 5

### ■ N-Body

- En física se usa para resolver problema de la predicción de los movimientos individuales de un grupo de objetos celestes que interactúan entre sí gravitacionalmente

#### Cálculo de fuerzas gravitatorias

$$F_{i,j} = m_i \frac{\partial^2 q_i}{\partial^2 t}$$

$$F_{i,j} = \frac{Gm_i m_j (q_j - q_i)}{\|q_j - q_i\|^3}$$

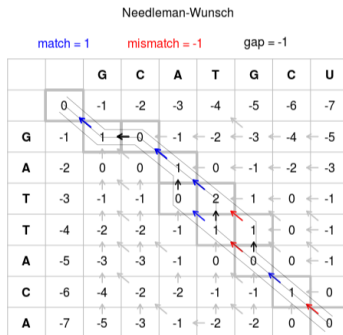
$$m_i \frac{\partial^2 q_i}{\partial^2 t} = \sum_{j=0, i \neq j}^N \frac{Gm_i m_j (q_j - q_i)}{\|q_j - q_i\|^3} = \frac{\partial U}{\partial q_i}$$



## Ejemplo 6

### ■ Needleman-Wunsch

- Algoritmo de alineamiento de secuencias
- Mejor alineamiento entre secuencias GCATGCU y GATTACA



## Ejemplo 6

- Needleman-Wunsch
- **PROBLEMA:** Dependencias de datos



## Ejemplo 6

```
1: procedure NW(a, b, S, d)
2:   for i = 0 to length(a) do
3:      $F_{i,0} \leftarrow d * i$ 
4:   end for
5:   for j = 0 to length(b) do
6:      $F_{0,j} \leftarrow d * j$ 
7:   end for
8:   for i = 0 to length(a) do
9:     for j = 0 to length(b) do
10:       $match \leftarrow F_{i-1,j-1} + S_{a_i,b_j}$ 
11:       $delete \leftarrow F_{i-1,j} + d$ 
12:       $insert \leftarrow F_{i,j-1} + d$ 
13:       $F_{i,j} \leftarrow \max(match, delete, insert)$ 
14:    end for
15:  end for
16: end procedure
```

▷ *a*, *b* son secuencias

