Arquitecturas Especializadas

Práctica 3. Programación con directivas

Carlos García Sánchez

Objetivos

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

OpenACC (I)

- Uso del compilador de PGI que soporta OpenACC
- GPU: nVIDIA Tesla M2070
  - Conocer características de la GPU

```
/opt/cuda-6.5/samples/1_Utilities/deviceQuery/deviceQuery
```

  - Wiki: http://en.wikipedia.org/wiki/CUDA

- Ejecución en gpuser@urbion.dacya.ucm.es
- Compilación en gpuser@speedy.dacya.ucm.es
  - Compilador de PGI

```
pgcc -Minfo -acc  hello.c -o hello.pgi.exe
```

  - Preparados makefiles:
    - CPU: **make host → bin.host.exe**
    - GPU: **make pgi → bin.pgi.exe**

Ejemplo 0

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

#ifdef _OPENACC
    acc_init(acc_device_r
    printf(" Compiling wi
#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;
}
```

```
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
```

Ejemplo 1

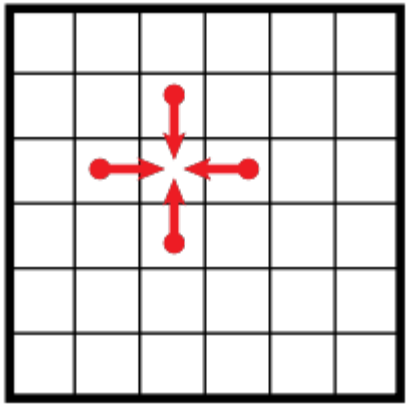
- Operación AXPY

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

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

  ....
  for(i=0; i<n; i++)
        y_acc[i] = a*x_acc[i] + y_acc[i];

   ...
#ifdef _OPENACC
    acc_shutdown(acc_device_not_host);
 #endif
  return 0;
}
```

Ejemplo 2

- Jacobi

Dependencias

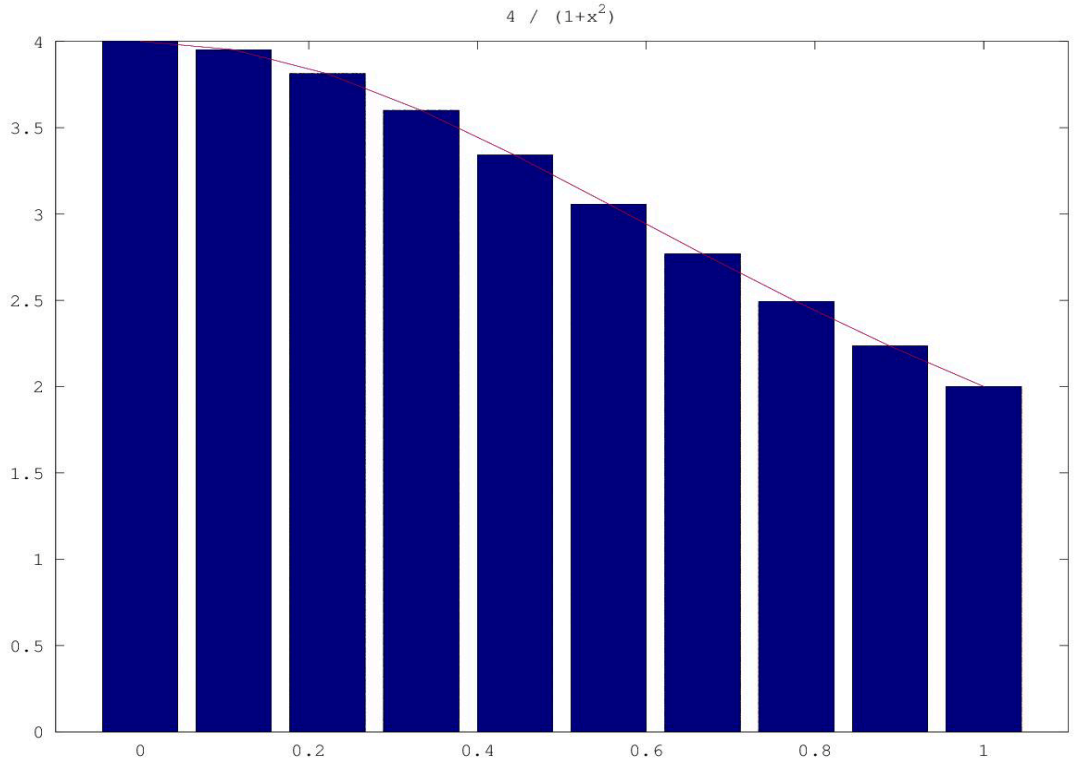


```
while ( error > tol && iter < iter_max ){
    error = 0.0;
    for( int j = 1; j < n-1; j++){
       for( int i = 1; i < m-1; i++ ){
          Anew[j][i] = 0.25 * ( A[j][i+1] + A[j][i-1]
                              + A[j-1][i] + A[j+1][i]);
          error = fmax( error, fabs(Anew[j][i] - A[j][i]));
       }
    }
    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++;
 }
```

Ejemplo 3

- Cálculo de PI

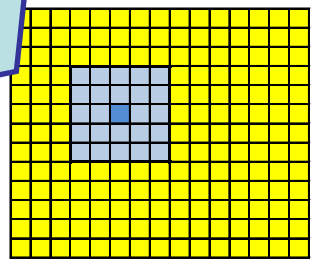
$$\prod = \int_0^1 \frac{4}{1+x^2} dx$$



Ejemplo 4

- Detección bordes→ convolucion
  - **./imgprocess input_image.bmp output_image_bmp [c,g]**
    - c: código ejecutado en CPU
    - g: código ejecutado en CPU

- Filtrado: convolución 2D



$$filt = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$