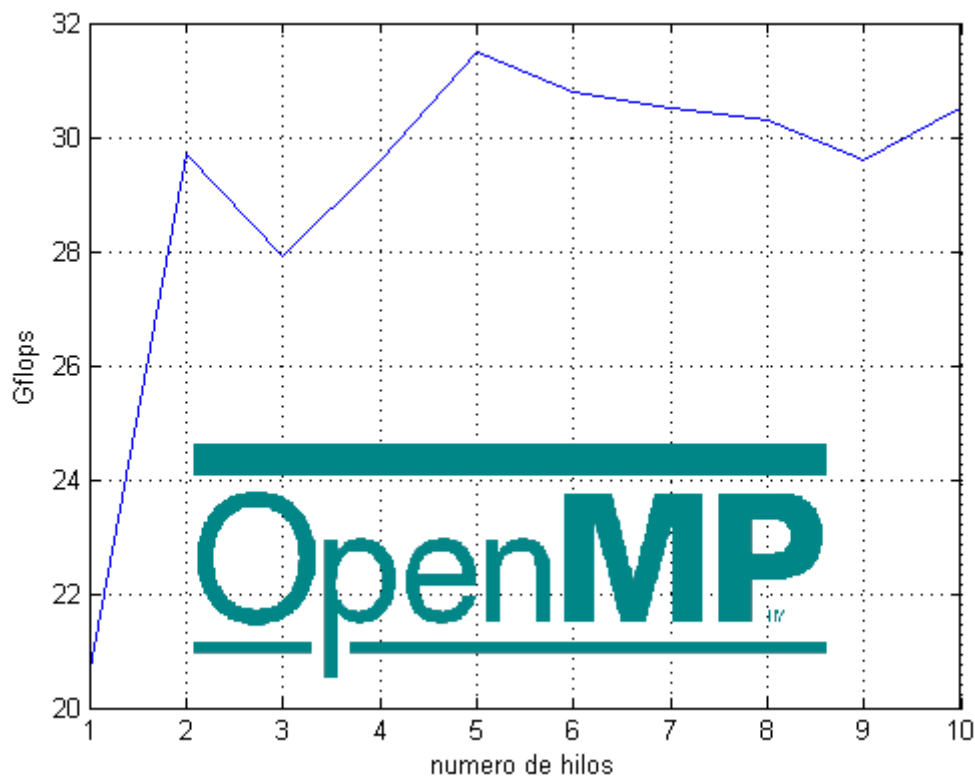


Práctica 6

Programación con OpenMP



Héctor Garbisu Arocha

Curso 2015/16

Métodos Numéricos para la Computación

Grado en Ingeniería Informática

Escuela de Ingeniería Informática

Universidad de Las Palmas de Gran Canaria

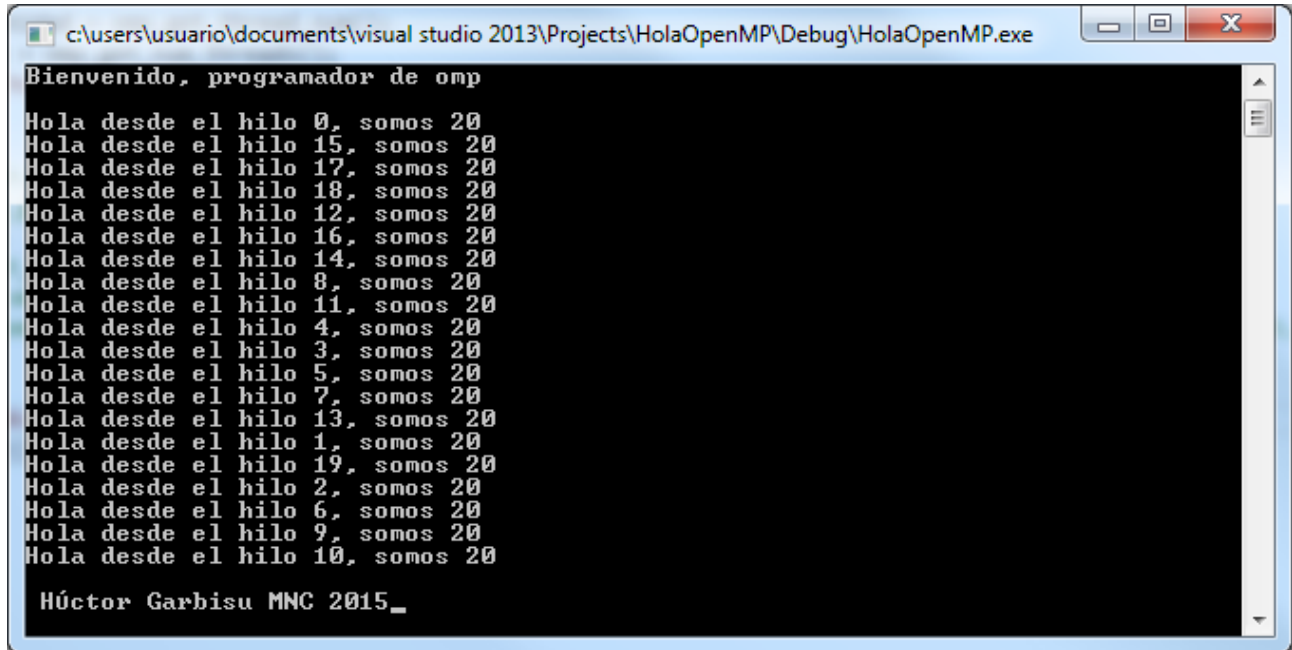
Índice

1. Hola OpenMP	pág. 3
2. OpenMP + BLAS	pág. 3
3. Uso de barrier y sections	pág. 5

1. Tarea 1. Hola OpenMP

En el primer ejercicio vamos a hacer un programa que se divida en varios hilos. Para comprobar que ésto se ha conseguido se imprimen los identificadores numéricos de los hilos.

Indicamos con una directiva de OMP que queremos que ese segmento del programa se ejecute paralelamente, con una función obtenemos el identificador de cada hilo. Primero hay que activar la compatibilidad con OMP desde las propiedades del proyecto.



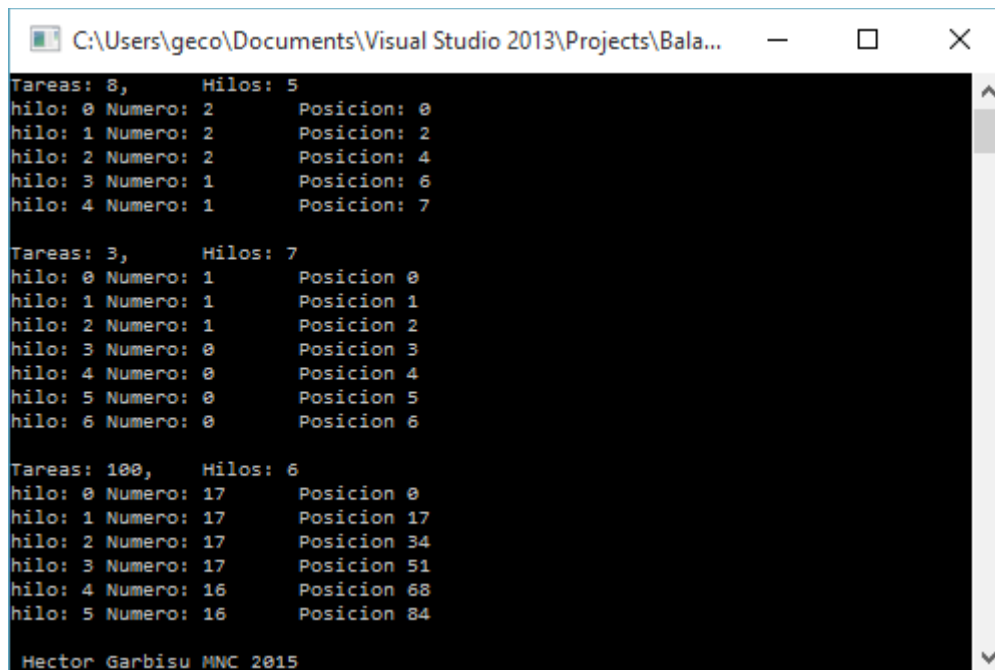
```
c:\users\usuario\documents\visual studio 2013\Projects\HolaOpenMP\Debug\HolaOpenMP.exe
Bienvenido, programador de omp
Hola desde el hilo 0, somos 20
Hola desde el hilo 15, somos 20
Hola desde el hilo 17, somos 20
Hola desde el hilo 18, somos 20
Hola desde el hilo 12, somos 20
Hola desde el hilo 16, somos 20
Hola desde el hilo 14, somos 20
Hola desde el hilo 8, somos 20
Hola desde el hilo 11, somos 20
Hola desde el hilo 4, somos 20
Hola desde el hilo 3, somos 20
Hola desde el hilo 5, somos 20
Hola desde el hilo 7, somos 20
Hola desde el hilo 13, somos 20
Hola desde el hilo 1, somos 20
Hola desde el hilo 19, somos 20
Hola desde el hilo 2, somos 20
Hola desde el hilo 6, somos 20
Hola desde el hilo 9, somos 20
Hola desde el hilo 10, somos 20

Húctor Garbisu MNC 2015_
```

2. Tarea 2. OpenMP + BLAS

Al igual que en prácticas anteriores, tenemos que activar MKL desde las propiedades del proyecto. Para no interferir con OMP de formas imprevistas, especificamos que sea en modo secuencial. La función BalanceoCarga se ocupa de hacer que cada hilo haga más o menos el mismo trabajo, con una diferencia máxima de una tarea entre diferentes hilos.

Podemos comprobar que BalanceoCarga distribuye las tareas entre los hilos ejecutando la demo:



```
Tareas: 8,      Hilos: 5
hilo: 0 Numero: 2      Posicion: 0
hilo: 1 Numero: 2      Posicion: 2
hilo: 2 Numero: 2      Posicion: 4
hilo: 3 Numero: 1      Posicion: 6
hilo: 4 Numero: 1      Posicion: 7

Tareas: 3,      Hilos: 7
hilo: 0 Numero: 1      Posicion 0
hilo: 1 Numero: 1      Posicion 1
hilo: 2 Numero: 1      Posicion 2
hilo: 3 Numero: 0      Posicion 3
hilo: 4 Numero: 0      Posicion 4
hilo: 5 Numero: 0      Posicion 5
hilo: 6 Numero: 0      Posicion 6

Tareas: 100,    Hilos: 6
hilo: 0 Numero: 17     Posicion 0
hilo: 1 Numero: 17     Posicion 17
hilo: 2 Numero: 17     Posicion 34
hilo: 3 Numero: 17     Posicion 51
hilo: 4 Numero: 16     Posicion 68
hilo: 5 Numero: 16     Posicion 84

Hector Garbisu MNC 2015
```

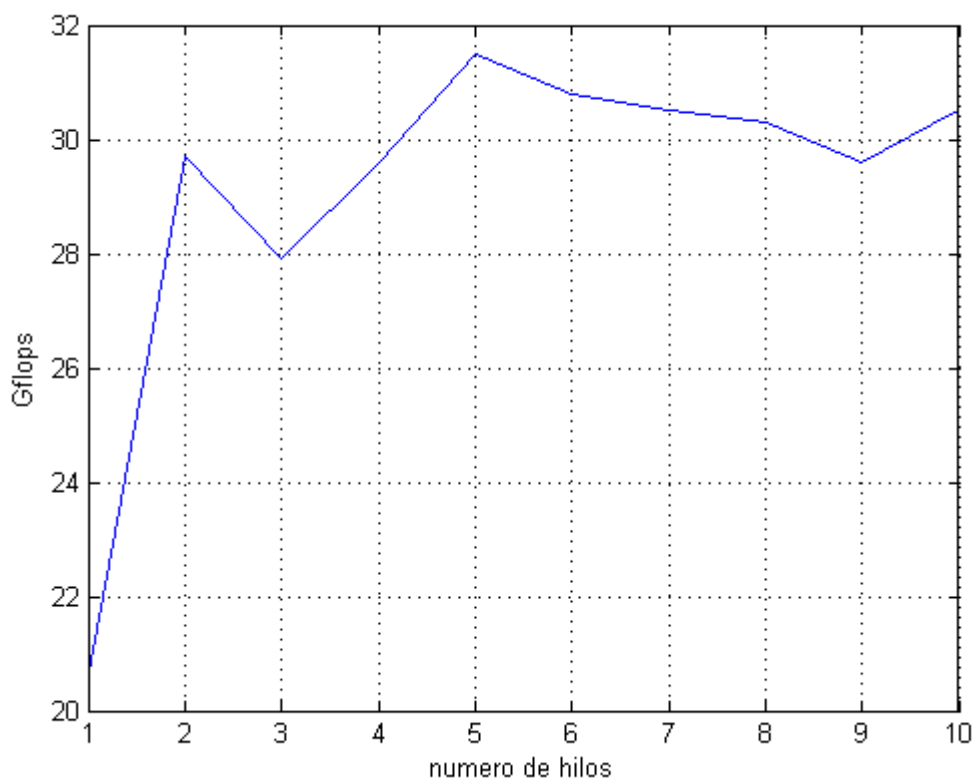
Para hacer pruebas sobre la eficiencia, el objetivo común que los hilos tendrán que hacer es multiplicar dos matrices cuadradas de 4096 filas. Cada hilo tendrá que hacer entonces $4096/\text{numhilos}$ multiplicaciones de una fila por la segunda matriz.

En cada ejecución de la prueba se evalúan diferentes valores de número de hilos, de 1 hasta nmuestras. En cada iteración se hace la misma multiplicación en paralelo cierto número de veces con la intención de garantizar que el resultado es consistente el tiempo total se divide por el número de repeticiones para obtener la media.

Este es el resultado para $\text{nmuestras} = 10$ y $\text{nrepeticiones} = 10$.

```
C:\Users\geco\Documents\Visual Studio 2013\Projects\Big...
Nthread: 1      Tiempo: 6.67757 s      Gflops: 20.5822
Nthread: 2      Tiempo: 4.63141 s      Gflops: 29.6754
Nthread: 3      Tiempo: 4.9243 s      Gflops: 27.9103
Nthread: 4      Tiempo: 4.64036 s      Gflops: 29.6181
Nthread: 5      Tiempo: 4.36955 s      Gflops: 31.4538
Nthread: 6      Tiempo: 4.4689 s      Gflops: 30.7546
Nthread: 7      Tiempo: 4.50791 s      Gflops: 30.4884
Nthread: 8      Tiempo: 4.53584 s      Gflops: 30.3007
Nthread: 9      Tiempo: 4.64193 s      Gflops: 29.6082
Nthread: 10     Tiempo: 4.50717 s      Gflops: 30.4934

Hector Garbisu MNC 2015_
```



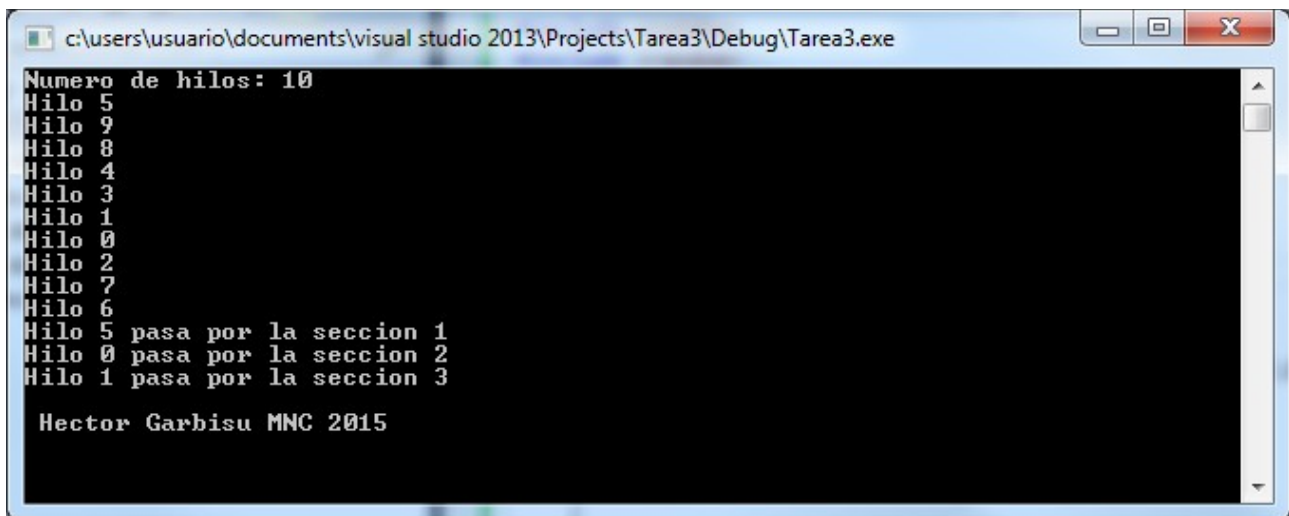
3. Tarea 3. Uso de barrier y sections

Como última tarea, utilizaremos la herramientas barrera y section para sincronizar varios hilos.

En el primer ejemplo, una barrera afecta a todos los hilos de modo que tienen que esperar a que termine el más lento (probablemente el 0 porque tiene una operación de e/s).

Tras eso todos los hilos reanudan su ejecución.

La segunda parte consiste en establecer 3 secciones. Las secciones también se implementan mediante directivas y se encargan fácilmente de hacer que por cada sección sólo pase un hilo, presumiblemente el primero que llega.



```
c:\users\usuario\documents\visual studio 2013\Projects\Tarea3\Debug\Tarea3.exe
Numero de hilos: 10
Hilo 5
Hilo 9
Hilo 8
Hilo 4
Hilo 3
Hilo 1
Hilo 0
Hilo 2
Hilo 7
Hilo 6
Hilo 5 pasa por la seccion 1
Hilo 0 pasa por la seccion 2
Hilo 1 pasa por la seccion 3

Hector Garbisu MNC 2015
```

BigMulr.cpp

```
#include <cstdio>
#include <random>
#include <mkl.h>
#include <omp.h>
#define Nbig 4*1024
#define MAXTHREADS 20
extern void BalanceoCarga(int Nth, int M, int *Pos, int *Num);
int main(int argc, char *argv[]){
    int Pos[MAXTHREADS], Num[MAXTHREADS];
    int Nth, N;
    double *A, *B, *C;
    bool test = false;
    int nmuestras = 10;
    int nrepeticiones = 10;
    N = Nbig;
    A = (double *)mkl_malloc(N*N*sizeof(double), 64);
    B = (double *)mkl_malloc(N*N*sizeof(double), 64);
    C = (double *)mkl_malloc(N*N*sizeof(double), 64);
    for (int muestra = 0; muestra < nmuestras; muestra++){
        std::default_random_engine generador;
        std::normal_distribution<double> distribucion(0.0, 1.0);
        for (int i = 0; i < N*N; i++){
            A[i] = distribucion(generador);
            B[i] = distribucion(generador);
        }
        Nth = muestra+1;
        BalanceoCarga(Nth, N, Pos, Num);
        int i;
        double inicio = omp_get_wtime();
#pragma omp parallel for private(i) num_threads(Nth)
        for (int repeticion = 0; repeticion < nrepeticiones; repeticion++){
            for (i = 0; i < Nth; i++){
                cblas_dgemm(CblasRowMajor, CblasNoTrans, CblasNoTrans,
Num[i], N, N, 1.0, &(A[Pos[i] * N]), N, B, N, 0.0, &(C[Pos[i] * N]), N);
            }
        }
        double fin = omp_get_wtime();
        double tiempo = fin - inicio;
        tiempo = tiempo / nrepeticiones;
        double Gflops = 2.0*N*N*N*1.0e-09 / tiempo;
        printf("\nNthread: %d\tTiempo: %g s\tGflops: %g\n", Nth, tiempo, Gflops);
    }
    mkl_free(A);
    mkl_free(B);
    printf("\n Hector Garbisu MNC 2015");
    std::getchar();
    return 0;
}
```