MASTER SIANI ULPGC
Computación Paralela
OpenMP

Sergio Marrero Marrero
Universidad de Las Palmas de Gran Canaria

16/05/2016

Índice

1. Optimización basada en paralelismo multihilos.Openmp

1.1. Objetivos de la practica

1. Características del multiprocesador
2. Usar Openmp para paralelizar el código propuesto en clase.
3. Ejecutar con diferente numero de hilos (1,2,4,8) manteniendo constante el número de operaciones.
4. Comentar los resultados

1.2. Tabla característica del multiprocesador:

- Nombre del modelo comercial: *Core i7-3610 QM*
- Año en el que se empezó a comercializar: *2012 (abril)*
- Microarquitectura del multiprocesador: *Ivy Bridge*
- Microarquitectura del Core:*Ivy Bridge (quad-core) (22 nm)*
- Tecnología:*22nm*
- Número de cores:*4*
- Número de instrucciones que se envían en paralelo(Issue Width):*2 (por core)*
- Tamaños de la memoria cache:*L1=4x64KB;L2=4x256KB;L3=6MB*

1.3. Paralelización del código propuesto

Se adjunta el código fuente con la paralelización correspondiente.

1.4. Desbordamiento de la memoria caché

Los distintos niveles de memoria caché tienen los siguientes tamaños:
L1=4x64KB;L2=4x256KB;L3=6MB

Sumando la contribución de los distintos niveles se tiene una cantidad total de 7.25 Mb. Suponiendo en todo momento matrices cuadradas, veamos que valor teórico n empieza a necesitar espacio fuera de la caché. En bytes esto será igual a:

$$V_{cache} = 7{,}25 * 1024^2$$

Si igualamos el volumen de datos de la caché a la memoria ocupada por la multiplicación matriz por vector se tendrá:

$$V_{datos} = 8 * (n^2 + 2n) \tag{1}$$

en donde: n^2 es tamaño de la matriz, $2n$ es el tamaño de los dos vectores (el que multiplica y el que recibe el resultado), y 8 es el tamaño en bytes de los floats.

Sabiendo esto podremos calcular qué valor de n desbordará la memoria caché:

$$Vcache = Vdatos \Rightarrow 7{,}25 * 1024^2 = 8 * (n^2 + 2n)$$

de donde se deduce que el tamaño de desborde es:

$$n = 973{,}82$$

1.5. Volumen de datos de cada matriz

Utilizando la expresión [1] podemos calcular el volumen de datos que se manejará en cada multiplicación. La tabla [1] muestra el tamaño(Mbyte) que se utiliza en cada operación. Se observa como la matriz de tamaño $n = 1000$ ya desborda el capacidad de la caché.

n	Volumen(Mbyte)
10	0.001
100	0.078
1000	7.7
10000	763.1

Cuadro 1: Volumen de cada multiplicación

1.6. Número de operaciones

El número de operaciones en coma flotante se calcula de la siguiente forma:

$$Nops = loops * 2 * n^2 \tag{2}$$

donde *loops* representará el número de veces que se repetirá cada multiplicación.

Los distintos experimentos se realizarán manteniendo constantes el número de operaciones en coma flotante, exactamente:

$$Nops = 2{,}0 \times 10^8 \tag{3}$$

La tabla 2 muestra los *loops* que se realizarán frente al tamaño de las matrices que se usarán.

n	loops
10	10^8
100	10^6
1000	10^4
10000	10^2

Cuadro 2: Repeticiones para cada matriz

1.7. Dieseño del experimento y resultados

Cada fila de la tabla 2 se repitió 10 veces y se anotó el tiempo medio y los Mflops medios en ejecutar el programa.El nivel de optimización con el que se compiló el programa fue -O2

A continuación se muestran las tablas con los valores medios, que resultaron de la ejecución del programa con distintos hilos. La tabla [3] muestra

el tiempo medio y Mflops medios (10 experimentos para calcular la media) corriendo en un solo hilo. Con 2,4,8 respectivamente las tablas [4],[5],[6]

n	tiempo(s)	Mflops
10	10.173	1966.018
100	9.227	2167.52
1000	10.32	1928.005
10000	10.172	1966.147

Cuadro 3: Número de hilos=1

n	tiempo(s)	Mflops
10	13.332	1516.748
100	4.941	4049.937
1000	5.456	3667.372
10000	5.481.3	3650.076

Cuadro 4: Número de hilos=2

n	tiempo(s)	Mflops
10	20.546	986.586
100	4.945	4224.873
1000	4.757	4381.548
10000	5.488	4043.601

Cuadro 5: Número de hilos=4

n	tiempo(s)	Mflops
10	23.581	852.641
100	3.764	6075.828
1000	3.817	5695.714
10000	4.933	4219.427

Cuadro 6: Número de hilos=8

A continuación se grafican los resultados.

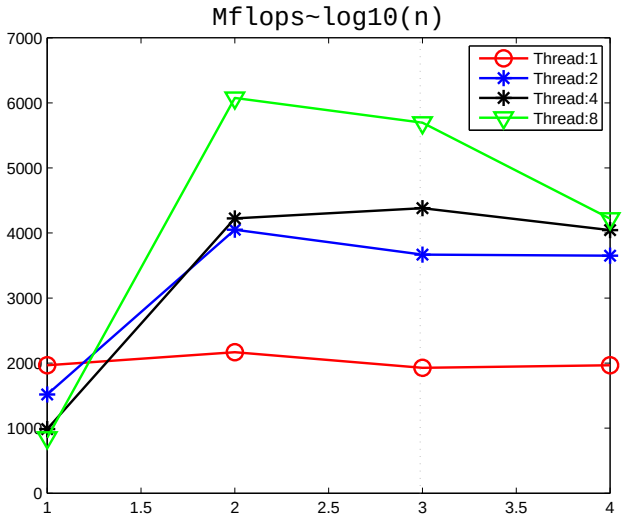

Figura 1: Tiempo de ejecucion frente a tamaño de las matrices

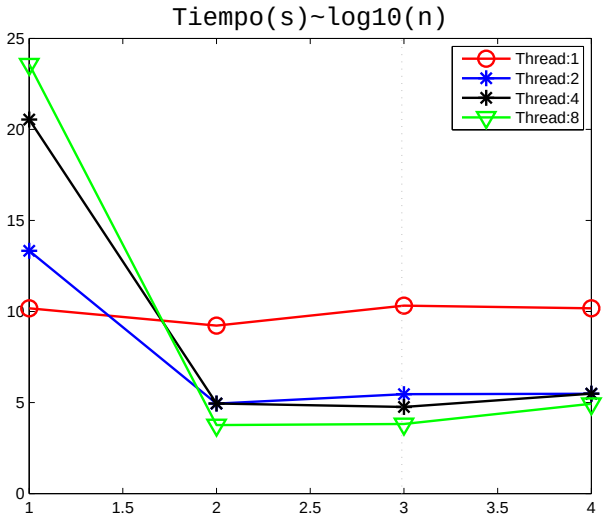

Figura 2: Tiempo de ejecucion frente a tamaño de las matrices

El la figura [1] se representa las distintas velocidades de ejecución de operaciones (Mflops) para los distintos hilos (ver leyenda). El la figura [2] se representa los distintos tiempos de ejecución para los distintos hilos (ver leyenda).

En ambas gráficas se ha representado con una linea punteada vertical de

color gris, la cual representa el tamaño a partir del cual la memoria requerida desborda la memoria caché.

1.8. Observaciones

- En general se observa una mejora en el rendimiento a medida que se aumentan los hilos.
- Para matrices de 10 hay un empeoramiento en el rendimiento conforme el número de hilos.
- No mejoran mucho las condiciones entre dos hilos y cuatro hilos.
- El desbordamiento de la caché se nota especialmente cuando se ejecuta en ocho hilos.