

Informe

Joaquin Villalba, Aldo Vizcaino

May 13, 2014

Contents

1	Informe	1
1.1	Original	2
2	Correccion	2
2.1	Dinámica	2
2.2	Estatica	2
2.3	Guiada	3
3	Tablas	3
3.1	OMP 4 hilos original	3
3.2	OMP 4 hilos corregido - dinamica	3
3.3	OMP 4 hilos corregido - estatica	4
3.4	OMP 4 hilos corregido - guiada	4
4	Conclusión	4
5	Hardware	4

1 Informe

El ejercicio tiene el problema de que el trabajo no está distribuido de forma uniforme entre los threads.

El algoritmo esta programado de forma tal que la fila $i+1$ procesa menos datos que la fila i . Por ejemplo, cuando el indice de filas sea 0, realizará el intercambio de posición en N elementos, y cuando el indice de filas sea $N-1$ sólo se realizará el intercambio de posiciones en dos elementos de la matriz.

1.1 Original

Por defecto se utiliza un schedule estatico, con un valor de chunk igual a numero de filas a procesar dividido cantidad de threads, entonces el thread 0 procesara de la fila 0 hasta la fila N/N_{Threads} .

Si $N = 8$ y se utilizan 2 threads, el scheduling resultante sera:

i	0	1	2	3	4	5	6	7
thread	1	1	1	1	2	2	2	2

Por esto, la carga de este thread es superior a la carga del ultimo thread creado, entonces, el scheduling por defecto no es el más indicado para utilizar.

2 Correccion

Para solucionar este inconveniente se utiliza la clausula schedule de OpenMP. Está directiva permite modificar como se divide el trabajo entre los threads. Esta distribución se puede llevar a cabo de diferentes maneras: Estática, Dinámica, Guiada, En ejecución y Automático.

Consideramos evaluar la clausula Dinámica, Estática y Guiada.

2.1 Dinámica

Esta clausula permite que cada thread luego de finalizar su correspondiente chunk comience a procesar el proximo chunk pendiente de ejecución. Es decir que a priori no se puede conocer que chunk ejecutará cada thread.

Sólo para ejemplificar podría ocurrir lo siguiente.

Continuando con el ejemplo: Si $N = 8$ y se utilizan 2 threads, el scheduling (static,1) resultante sera:

i	0	1	2	3	4	5	6	7
thread	1	2	2	1	1	1	2	1

nota: El chunk que utiliza por defecto cuando no es especificado por parametro es 1

2.2 Estatica

Con la clausula static podemos especificar que cantida de elemento del *for* serán procesados por un thread de manera consecutiva.

Continuando con el ejemplo: Si $N = 8$ y se utilizan 2 threads, el scheduling (static,1) resultante sera:

i	0	1	2	3	4	5	6	7
thread	1	2	1	2	1	2	1	2

2.3 Guiada

La clausula guided es similar a la clausula dynamic, con la diferencia de que el procesamiento se agrupa en bloques y el tamaño del bloque decrece a medida que se asigna trabajo a un thread. Por defecto el tamaño inicial de bloque es N/N_{threads} .

Solo para ejemplificar podría ocurrir lo siguiente.

Continuando con el ejemplo: Si $N = 8$ y se utilizan 2 threads, el scheduling (guided,1) resultante sera:

i	0	1	2	3	4	5	6	7
thread	1	1	1	1	2	2	1	2

nota: En este caso el parametro que se le pasa al schedule significa el tamaño minimo de chunk que se le asignara a un thread.

3 Tablas

3.1 OMP 4 hilos original

N	Tiempo Secuencial	Tiempo	Speedup	Eficiencia
2048	0.068505	0.038079	1.7990231	0.44975578
4096	0.284406	0.157106	1.8102810	0.45257025
8192	1.163672	0.625998	1.8589069	0.46472673

Volver a original

3.2 OMP 4 hilos corregido - dinamica

N	Tiempo Secuencial	Tiempo	Speedup	Eficiencia
2048	0.068505	0.019999	3.4254213	0.85635533
4096	0.284406	0.093363	3.0462389	0.76155973
8192	1.163672	0.427885	2.7195905	0.67989763

Volver a dinamica

3.3 OMP 4 hilos corregido - estatica

N	Tiempo Secuencial	Tiempo	Speedup	Eficiencia
2048	0.068505	0.020757	3.3003324	0.8250831
4096	0.284406	0.095250	2.9858898	0.74647245
8192	1.163672	0.433647	2.6834545	0.67086363

Volver a estatica

3.4 OMP 4 hilos corregido - guiada

N	Tiempo Secuencial	Tiempo	Speedup	Eficiencia
2048	0.068505	0.039383	1.7394561	0.43486403
4096	0.284406	0.159459	1.7835682	0.44589205
8192	1.163672	0.649765	1.7909121	0.44772803

Volver a guiada

4 Conclusión

La ejecucion con la clausula dynamic retorno mejores tiempos de eficiencia. Se observa que a medida que crece N la eficiencia disminuye. Esta baja podria ser compensada incrementando el tamaño de los chunks, pero, se observo que en ese caso bajo la eficiencia para los N más chicos.

5 Hardware

- Se utilizaron las máquinas de la sala de postgrado.
- Para compilar todos los archivos se adjunta un archivo Makefile.