

Informe sobre multiplicacion matriz vector y ordenamiento odd even sort en openmp

Luis Alberto Sánchez Moreno Colque

October 2, 2018

1 Ordenamiento Odd Even Sort

A continuación se muestran los datos del libro con two parallel for y two for.
[1].

Table 5.2 Odd-Even Sort with Two parallel for Directives and Two for Directives (times are in seconds)				
thread.count	1	2	3	4
Two parallel for directives	0.770	0.453	0.358	0.305
Two for directives	0.732	0.376	0.294	0.239

Aquí se muestran los tiempos obtenidos en nuestra pc

thread count	1	2	3	4
Two parallel for	1.23	0.95	0.68	0.41
Two for	0.094	0.99	0.53	0.35

2 Multiplicación de Matriz Vector

En el siguiente cuadro se muestra los resultados del libro.

Table 5.4 Run-Times and Efficiencies of Matrix-Vector Multiplication (times in seconds)						
Threads	Matrix Dimension					
	8,000,000 × 8		8000 × 8000		8 × 8,000,000	
	Time	Eff.	Time	Eff.	Time	Eff.
1	0.322	1.000	0.264	1.000	0.333	1.000
2	0.219	0.735	0.189	0.698	0.300	0.555
4	0.141	0.571	0.119	0.555	0.303	0.275

El sistema de 8000000x8 requiere aproximadamente más tiempo que el sistema 8000x8000, y el sistema de 8x8000000 requiere más de tiempo que el sistema 8000x8000. Ambas diferencias se pueden atribuir al menos parcialmente al rendimiento de la memoria caché. Sin embargo, las diferencias se nota en la eficiencia a medida que aumenta el número de threads. La eficiencia de dos hilos del programa con la entrada de 8x8000000 es menor que la eficiencia del programa con las entradas 8000000x8 y 8000x8000. La eficiencia de cuatro hilos del programa con la entrada de 8x8000000 es super menor que la eficiencia del programa con las entradas 8000000x8 y 8000x8000.

Implementación	8000000x8	2	8000x8000	8	8x8000000	
threads	tiempo	efi	tiempo	efi	tiempo	efi
1	0.78	1.00	0.65	1.00	0.88	1.00
2	0.52	0.75	0.48	0.67	0.72	0.61
4	0.46	0.56	0.40	0.54	0.69	0.43

References

- [1] Book. an introduction to parallel programming, nov 2011.