Análisis OpenMP

Wilber Cutire

April 24, 2018

De acuerdo al Cápitulo 4 de Computación Paralela en [1] podemos analizar y implementar la paralelización de estructuras y algoritmos mediante pThreads. En este trabajo reproduciremos la Tablas 4.1 donde a través de una función que acumule en el calculo de pi se compara las implementaciones usando busy-wait y mytex, la Tabla 4.3 y 4.4 donde analizamos la implementación de la sección crítica de una lista enlazada con candados read/write, un mutex para toda la lista y un mutex por cada nodo, y por último también la Tabla 4.5 para la implementaión de la multiplicación matriz-vector usando pThreads.

Table 1: Busy-Wait vs Mutex

Threads	Busy-Wait	Mutex
1	3.974	3.927
2	2.115	2.055
4	1.340	1.229
8	1.131	1.072
16	1.259	1.1035
32	2.052	1.083
64	5.292	1.156

Table 2: Tiempo de operaciones de Linked List: 99.9% busqueda, 0.05% inserciones, 0.05% eliminaciones

Threads	Read-Write Locks	1 Mutex	n-Mutex
1	2.016	2.188	1.106
2	1.051	9.287	9.624
4	5.876	9.401	6.075
8	5.149	9.736	8.371

Table 3: Tiempo de operaciones de Linked List: 80.0% busqueda, 10.0% inserciones, 10.0% eliminaciones

Threads	Read-Write Locks	1 Mutex	n-Mutex
1	17.054	16.531	58.378
2	13.767	26.868	46.762
4	11.228	27.598	38.887
8	11.709	28.175	29.909

Table 4: Tiempos de Multiplicación Matriz-Vector OpenMP

Threads	8000000x8	8000x8000	8x8000000
1	2.408	2.327	2.157
2	1.206	1.026	1.266
4	7.794	6.280	7.742

1 Conclusión

De acuerdo a los resultados, la eficiencia de la paralelización de un programa es dependiente de los cores en este caso usamos un procesador i7 con 8 cores, por lo que si no se distribuye bien los threads puede saturarse la comunicación y por tanto perder rendimiento. Por ejemplo mientras más inserciones hagamos a una lista enlazada las dependencias de comunicación incrementan y por tanto en práctica la eficiencia se ve afectada por la dependencia de comunicación en la sección crítica, y se aleja de complejidad teorica del programa.

References

[1] Peter Pacheco. *An introduction to parallel programming.* Elsevier, 2011.