

Pérez_Rubén_Tarea3

May 6, 2022

1 Computo Paralelo

2 Tarea 1

Rubén Pérez Palacios Profesor: Dr. Luis Daniel Blanco Cocom 5 de Mayo del 2022

2.1 Introducción

Se generó \$ 10000 \$ números enteros en el intervalo $[0, 50]$, con el código *random_num*. Posteriormente con esos números se ejecuto y medio el tiempo en microsegundos de los siguientes códigos:

- bubble_sort.cpp,
- odd-even_transposition_sort,
- shell_sort(parallel),
- shell_sort(serial).

Esto promediando el tiempo de 10 ejecuciones de cada uno para los resultados, y con 1, 2, 3, 4 hilos para los códigos en paralelo.

2.2 Explicación

De los códigos proporcionados del *BubbleSort* y *OddEvenTranspositionSort*, se implementaron en *c++*. Se uso el método *ShellSort* para ordenar un arreglo *num*, el cual consiste en ir iterando sobre una lista *h* de tamaños (normalmente decreciente) que contenga el 1, para cada tamaño se extraen los subarreglos $num_{h,k}$, donde todos los elementos de $num_{h,k}$ en el arreglo *num* están en posiciones con residuo *k* módulo *h*, cada uno de esos arreglos se ordena. El que la lista de tamaños contenga 1 asegura que el arreglo termine ordenado. Se prefiere ir iterando la lista de tamaños de manera decreciente para que cada vez que se orden los subarreglos se tenga que hacer menos intercambios.

Para compilar los códigos se uso los siguientes comandos:

- Compilar serial: `g++ -DNDEBUG -g file_name.cpp -o file_name`
- Compilar paralelo: `g++ -DNDEBUG -g -fopenmp file_name.cpp -o file_name`

Para ejecutar en cmd de windows los programas compilados se uso el siguiente comando:

- `file_name.exe < file_name_1.txt`

Donde “file_name_1.txt” es un archivo que contiene el tamaño del arreglo y los valores de este.

2.3 Tiempos

Los tiempos obtenidos para cada uno de los códigos son:

Bubble : 338306

Hilos	OETS	Shell
1	193615	2532
2	189274	2171
3	189815	2399
4	181632	2042

Donde OETS = “odd-even_transposition_sort”. Se puede ver que el bubble sort es el peor aún cuando los otros métodos se corrieron en 1 hilo, por su complejidad cuadrática. El OETS tiene una mejora considerable contra el Bubble sort. El shell sort al tener una complejidad promedio $O(N\log N)$ tiene el mejor tiempo.