

Ejercicios 5.3

Luis Tong

May 2017

Ejercicios 5.3 OpenMP Counting Sort

a. Si intentamos para paralelizar el bucle (el bucle externo), qué variable debería ser privada y qué se debería compartir?

Las variables ‘i’, ‘j’ y ‘count’ deben ser privadas, mientras que las variables ‘a’, ‘n’ y ‘temp’ deben ser compartidas.

b. Si se paralela el bucle que utiliza el scoping que se especificó en la parte anterior, hay alguna dependencia de bucle dependiente? Explica tu respuesta..

No habrá dependencia de bucle, ya que ninguna variable compartida se cambia en la misma posición por dos subprocesos. Veamos, las variables ‘a’ y ‘n’ nunca se escriben, por lo que no hay conflictos en ella.

La variable ‘temp’ es modificada por los hilos, pero como cada elemento del vector tiene exactamente una posición bien definida, ningún subproceso va a escribir en la misma posición que otra.

c. ¿Podemos igualar la llamada a memcpy? ¿Puedo modificar el código para que esta parte de la función será paralela?

La llamada a ‘memcpy’ no puede ser paralelizada, sin embargo podemos cambiar el código para realizar esta operación en un bucle; En este caso podemos paralelizarlo.

```
# pragma omp parallel for shared(a, n, temp) private(i)
for (i = 0; i < n; i++)
a[i] = temp[i];
```

d. Escribir un programa en C que incluye una combinación paralela de una orden de puntuación

El programa está en ‘main.c’.

e. Cómo realizar el rendimiento de su paralelización de la cuenta de grupo Count sort? ¿Cómo se compara con la serie de la biblioteca de archivos qsort?

El algoritmo de counting-sort dado es cuadrático en el número de elementos a ser ordenados. Las pruebas mostraron que, aun siendo paralelizado, el algoritmo pierde para el quick-sort, que es ‘ $n \log n$ ’. La siguiente tabla muestra los resultados generados al realizar el benchmark.

Elements	1-Thread	2-Threads	3-Threads	4-Threads	Quick-Sort
1000	9160108	4669623	3164560	2956490	230942
10000	915967266	473209914	310594737	244862219	1882896
100000	91623892739	45809274943	30736720102	24242175754	22359816

Ahora se muestra el speedup en relación al Counting-Sort con un subproceso:

Elements	1-Thread	2-Threads	3-Threads	4-Threads	Quick-Sort
1000	1.0	1.96	2.89	3.1	39.66
10000	1.0	1.94	2.95	3.74	486.47
100000	1.0	2.00	2.98	3.78	4097.70