

TestBP3.pdf



Sanchez01



Arquitectura de Computadores



2º Grado en Ingeniería Informática



Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
Universidad de Granada



#ESTASREADYCOLACAO

ColaCao



1. ¿Cuál de las siguientes formas es la correcta para fijar a 4 el número de hebras para un programa OpenMP ?
 - a. En un programa OpenMP, usando la función `omp_max_threads(4)` al principio de la función `main`.
 - b. En un programa OpenMP, usando la función `omp_num_threads(4)` al principio de la función `main`.
 - c. En la consola del sistema, usando la variable de entorno `export OMP_THREAD_LIMIT=4`
 - d. En un programa OpenMP, usando la función `omp_set_num_threads(4)` al principio de la función `main`.

2. Indica cuál de las siguiente opciones obtendrá mejores prestaciones para multiplicar una matriz triangular por un vector.
 - a.

```
#pragma omp for private(j)
schedule(guided)
for (i=0 ; i<N ; i++) {
    v2[i] = 0;
    for (j=0 ; j<=i ; j++) {
        #pragma omp critical
        v2[i] += M[i][j] * v1[j];
    }
}
```
 - b.

```
#pragma omp for private(j) schedule(static)
for (i=0 ; i<N ; i++) {
    v2[i] = 0;
    for (j=0 ; j<N ; j++)
        v2[i] += M[i][j] * v1[j];
}
```
 - c.

```
#pragma omp for private(j) schedule(guided)
for(i=0 ; i<N ; i++) {
    v2[i] = 0;
    for (j=0 ; j<=i ; j++)
        v2[i] += M[i][j] * v1[j];
}
```
 - d.

```
#pragma omp for schedule(guided)
for (i=0 ; i<N ; i++) {
    v2[i] = 0;
    for (j=0 ; j<=i ; j++)
        v2[i] += M[i][j] * v1[j];
}
```

3. Dado el código que se tiene a continuación ¿Qué tipo de reparto de iteraciones a hebras sería el más óptimo en tiempo de ejecución?


```
#pragma omp parallel for
for (int i=0 ; i<100 ; i++)
    a[i] += b[i]*c[i];
```

 - a. `dynamic`
 - b. `guided`

- c. runtime
- d. static

4. Analiza el código mostrado a continuación e indica qué habría que cambiar para que se imprima la siguiente salida. Cuando `OMP_NUM_THREADS = 4`.

```
int i, n = 3;
#pragma omp parallel for private(n)
for (i = 0; i < omp_get_max_threads(); ++i)
    printf("Thread %d imprime: %d", omp_get_thread_num(), i+n);
```

Salida por pantalla:

```
Thread 0 imprime: 3
Thread 1 imprime: 4
Thread 2 imprime: 5
Thread 3 imprime: 6
```

- a. Cambiar `private` por `copyprivate`
 - b. No hay que cambiar nada
 - c. Cambiar `private` por `firstprivate`
 - d. Cambiar `private` por `lastprivate`
5. El tiempo de ejecución de un programa paralelo ...
- a. Puede ser mayor que el tiempo de la versión secuencial para tamaños de problema pequeños, debido a la sobrecarga introducida al crear y destruir las hebras.
 - b. Siempre será menor que el de su versión secuencial, para cualquier tamaño del problema.
 - c. Aumenta conforme el tamaño del problema disminuye.
 - d. Se reduce conforme el tamaño del problema aumenta.

6. En el siguiente fragmento de código, ¿cuántas hebras están ejecutando la región paralela?

```
long sum = 0, N=10, a[10], b[10], c[10];
#pragma omp parallel
{
    int ithread = omp_get_thread_num();
    int nthread = omp_get_num_threads();
    #pragma omp sections
    {
        #pragma omp section
        for (long i = 0; i < N; i += nthread)
            c[i] = a[i] + b[i];
        #pragma omp section
        for (long i = ithread; i < N; i += nthread)
            c[i] = a[i] + b[i];
    }
}
```

- a. Las que indique la función `omp_get_thread_num()`
- b. El número de hebras posible será siempre igual al número de procesadores lógicos que tenga la máquina donde se ejecuta el código.
- c. N
- d. 2

7. Las variables de control internas de OpenMP

- a. Sólo se pueden modificar mediante el uso de variables de entorno en la consola del sistema
- b. Pueden ser accedidas directamente por el programador
- c. Las otras respuestas son todas incorrectas
- d. Sólo se pueden modificar mediante el uso de las funciones que proporciona el API de OpenMP

8. Indica qué reparto de iteraciones a hebras es correcto suponiendo 2 hebras y la cláusula `schedule(guided,3)`

a)

iteración	0	1	2	3	4	5	6	7	8	9	10	11
hebra	0	0	0	0	0	0	1	1	1	0	0	0

b)

iteración	0	1	2	3	4	5	6	7	8	9	10	11
hebra	0	0	0	0	0	0	1	1	1	0	0	1

c)

iteración	0	1	2	3	4	5	6	7	8	9	10	11
hebra	0	0	0	0	0	0	0	1	1	0	0	0

d)

iteración	0	1	2	3	4	5	6	7	8	9	10	11
hebra	0	0	0	1	1	1	0	0	0	1	1	1

9. Si le piden que realice un estudio de escalabilidad de un código que calcula el producto de dos matrices.

- a. Representaría en una gráfica el tiempo de ejecución en función del número de núcleos.
- b. Representaría en una gráfica la ganancia en velocidad (o ganancia en prestaciones) en función del número de núcleos.
- c. Representaría en una gráfica el tiempo de ejecución en función del tamaño de las matrices.
- d. No haría nada de lo indicado en el resto de respuestas.

10. ¿Qué valor toma la variable `kind` en el siguiente código?

```
#pragma omp parallel for schedule(static)
for (i=0; i<10000; i++){
    if (i==0)
        omp_get_schedule(&kind, &chunk);
    v3[i] = v1[i] + v2[i];
}
```

- a. `omp_sched_static`
- b. El que indique la variable de control interno `run-sched-var`.
- c. Ninguna otra respuesta es correcta.
- d. El que indique la variable de control interno `def-sched-var`.