

Hoja de Trabajo 1

Competencias a desarrollar:

Implementa y diseña programas para la paralelización de procesos con memoria compartida usando OpenMP.

1. Instrucciones

Resuelva los ejercicios hechos en clase, haciendo las modificaciones usando OpenMP para crear soluciones paralelas de un programa secuencial. Deje evidencia de todo su trabajo y realice los cálculos de speedup en donde se le indique. Recuerde que para calcular el speedup de un programa usando tiempo, es necesario que realice varias mediciones y use una de los siguientes: tiempo máximo o tiempo promedio.

La hoja puede hacerla individual o en equipos, hasta un máximo de 3 integrantes. Cada alumno debe entregar su copia individual de la hoja en formato PDF. Antes del siguiente día de clases.

En todos los incisos debe dejar copia de lo siguiente:

- Función(es) principal(es) del código. La parte que contiene el cálculo o tarea más importante. No incluya encabezados ni operaciones de interacción con el usuario.
 - Bloque de código modificado
- Captura de pantalla de la salida (x1)
- Mediciones de tiempo (min 5)
- Métrica de desempeño: speedup, efficiency

2. Programa secuencial

Escriba acá el código secuencial que modificará para convertirlo en paralelo mediante OpenMP. Compílelo y ejecútelo, registrando siempre el tiempo de ejecución del programa. (Realice varias mediciones y deje evidencia. Incluya como mínimo una captura de pantalla de la salida del programa)

```
integral = trapezoides(a,b,n);
```

```
double trapezoides(double a, double b, int n) {
    double integral, h;
    int k;

    //---- Ancho de cada trapezoide
    h = (b-a)/n;

    //---- Valor inicial de la integral (valores extremos)
    integral = (f(a) + f(b)) / 2.0;

    for(k = 1; k <= n-1; k++) {
        integral += f(a + k*h);
    }
    integral = integral * h;

    return integral;
}/*trapezoides*/
```

No.	Ejecución	Tiempo
1	PS C:\Users\jfdel\Desktop\UVG2022-1\Computacion Paralela y Distribuida\HDT1\Code> Con n = 10000000 trapezoides, nuestra aproximacion de la integral de 1.000000 a 40.000000 es = 85332.0000000067 command took 0:0:0.09 (0.09s total)	0.09
2	PS C:\Users\jfdel\Desktop\UVG2022-1\Computacion Paralela y Distribuida\HDT1\Code> Con n = 10000000 trapezoides, nuestra aproximacion de la integral de 1.000000 a 40.000000 es = 85332.0000000067 command took 0:0:0.08 (0.08s total)	0.08
3	PS C:\Users\jfdel\Desktop\UVG2022-1\Computacion Paralela y Distribuida\HDT1\Code> Con n = 10000000 trapezoides, nuestra aproximacion de la integral de 1.000000 a 40.000000 es = 85332.0000000067 command took 0:0:0.08 (0.08s total)	0.08
4	PS C:\Users\jfdel\Desktop\UVG2022-1\Computacion Paralela y Distribuida\HDT1\Code> Con n = 10000000 trapezoides, nuestra aproximacion de la integral de 1.000000 a 40.000000 es = 85332.0000000067 command took 0:0:0.08 (0.08s total)	0.08
5	PS C:\Users\jfdel\Desktop\UVG2022-1\Computacion Paralela y Distribuida\HDT1\Code> Con n = 10000000 trapezoides, nuestra aproximacion de la integral de 1.000000 a 40.000000 es = 85332.0000000067 command took 0:0:0.09 (0.09s total)	0.09

3. Secciones críticas

Acciones que realizó para probar el uso de directivas para manejar acceso controlado a memoria.

No.	Ejecución	Tiempo
1	PS C:\Users\jfdel\Desktop\UVG2022-1\Computacion Paralela y Distribuida\HDT1\Code> With n = 1000000 trapezoids, our estimate of the integral from 1.000000 to 40.000000 = 2.13330000000017e+004 command took 0:0:0.07 (0.07s total)	0.07
2	PS C:\Users\jfdel\Desktop\UVG2022-1\Computacion Paralela y Distribuida\HDT1\Code> With n = 1000000 trapezoids, our estimate of the integral from 1.000000 to 40.000000 = 2.13330000000017e+004 command took 0:0:0.08 (0.08s total)	0.08
3	PS C:\Users\jfdel\Desktop\UVG2022-1\Computacion Paralela y Distribuida\HDT1\Code> With n = 1000000 trapezoids, our estimate of the integral from 1.000000 to 40.000000 = 2.13330000000017e+004 command took 0:0:0.08 (0.08s total)	0.08
4	PS C:\Users\jfdel\Desktop\UVG2022-1\Computacion Paralela y Distribuida\HDT1\Code> With n = 1000000 trapezoids, our estimate of the integral from 1.000000 to 40.000000 = 2.13330000000017e+004 command took 0:0:0.07 (0.07s total)	0.07
5	PS C:\Users\jfdel\Desktop\UVG2022-1\Computacion Paralela y Distribuida\HDT1\Code> With n = 1000000 trapezoids, our estimate of the integral from 1.000000 to 40.000000 = 2.13330000000017e+004 command took 0:0:0.07 (0.07s total)	0.07

4. Operaciones de reducción

Acciones que realizó para comprobar el uso de operaciones de fusión.

No.	Ejecución	Tiempo
1	Ps C:\Users\jfdel\Desktop\UVG2022-1\Computacion Paralela y Distribuida\HDT1\Code> With n = 1000000 trapezoids, our estimate of the integral from 1.000000 es =21333.0000000001 real 0m0.009s user 0m0.011s sys 0m0.000s	0.009s
2	Ps C:\Users\jfdel\Desktop\UVG2022-1\Computacion Paralela y Distribuida\HDT1\Code> With n = 1000000 trapezoids, our estimate of the integral from 1.000000 es =21333.0000000001 real 0m0.013s user 0m0.104s sys 0m0.009s	0.013
3	Ps C:\Users\jfdel\Desktop\UVG2022-1\Computacion Paralela y Distribuida\HDT1\Code> With n = 1000000 trapezoids, our estimate of the integral from 1.000000 es =21333.0000000001 real 0m0.012s user 0m0.071s sys 0m0.000s	0.012s

4	<pre> Ps C:\Users\jfdel\Desktop\UVG2022-1\Computacion Paralela y Distribuida\HDT1\Code> With n = 1000000 trapezoids, our estimate of the integral from 1.000000 es =21333.0000000001 real 0m0.013s user 0m0.000s sys 0m0.020s </pre>	0.013s
5	<pre> Ps C:\Users\jfdel\Desktop\UVG2022-1\Computacion Paralela y Distribuida\HDT1\Code> With n = 1000000 trapezoids, our estimate of the integral from 1.000000 es =21333.0000000001 real 0m0.012s user 0m0.017s sys 0m0.000s </pre>	0.012s

T Max: 0.013s

T prom: 0.0118 s

Velocidad: 2.71

Eficiencia: 0.226

5. Loops paralelos

Acciones que realizó para comprobar el manejo y creación de loops FOR paralelos.

No.	Ejecución	Tiempo
1	<pre> PS C:\Users\jfdel\Desktop\UVG2022-1\Computacion Paralela y Distribuida\HDT1\Code> Con n = 1000000 trapezoides, nuestra aproximacion de la integral de 1.000000 a 40.000000 es = 85332.0000000067 command took 0:0:0.09 (0.09s total) </pre>	0.09
2	<pre> PS C:\Users\jfdel\Desktop\UVG2022-1\Computacion Paralela y Distribuida\HDT1\Code> Con n = 1000000 trapezoides, nuestra aproximacion de la integral de 1.000000 a 40.000000 es = 85332.0000000067 command took 0:0:0.07 (0.07s total) </pre>	0.07
3	<pre> PS C:\Users\jfdel\Desktop\UVG2022-1\Computacion Paralela y Distribuida\HDT1\Code> Con n = 1000000 trapezoides, nuestra aproximacion de la integral de 1.000000 a 40.000000 es = 85332.0000000067 command took 0:0:0.10 (0.10s total) </pre>	0.10
4	<pre> PS C:\Users\jfdel\Desktop\UVG2022-1\Computacion Paralela y Distribuida\HDT1\Code> Con n = 1000000 trapezoides, nuestra aproximacion de la integral de 1.000000 a 40.000000 es = 85332.0000000067 command took 0:0:0.08 (0.08s total) </pre>	0.08
5	<pre> PS C:\Users\jfdel\Desktop\UVG2022-1\Computacion Paralela y Distribuida\HDT1\Code> Con n = 1000000 trapezoides, nuestra aproximacion de la integral de 1.000000 a 40.000000 es = 85332.0000000067 command took 0:0:0.08 (0.08s total) </pre>	0.08

6. Cambio en el entorno de las variables

Acciones que realizó para comprobar el efecto que tiene el cambio del entorno (scope) de una variable en un bloque paralelo.

```
void prodAx(int m, int n, double * restrict A, double * restrict x,
double * restrict b){

    int i, j;
    #pragma omp parallel for shared(m,n,A,x,b) private(i,j) schedule(guided, 100000)

    for(i=0; i<m; i++){
        b[i]=0.0;

        for(j=0; j<n; j++){
            b[i] += A[i*n + j] * x[j];
        }
    }
}
```

```
amado-developer@Amados-MacBook-Pro:~
→ ~ clang -Xpreprocessor -fopenmp -o inciso6 prodAx.c -lomp
→ ~ ./inciso6
Ingrese las dimensiones m y n de la matriz: 10 10
Initializing matrix A and vector x
Calculando el producto Ax para m = 10 n = 10

b:
      221      353      229      173      193      177      224      215      219      2
11
→ ~
```

7. Ajustes en las formas de planificación (scheduling)

Acciones que realizó para comprobar el efecto que tienen las diferentes formas de scheduling sobre el desempeño de un programa paralelo.

```
>./ejercicio
Ingrese las dimensiones: 1000 1000
iniciando matriz
calculando producto m= 1000 n = 1000

B:
27702 27059 27714 27707 28421 27680 27647 27010 29130 27701 27701 27785
```

#####	D(segundos)	S(segundos)	G(segundos)
1000	14.87	15.01	17.22
10000	12.98	14.87	14.95
100000	11.84	14.94	13.66