

## Trabajo Nº 3

### Programación en pasaje de mensajes / Programación híbrida

---

**Fecha límite para el envío: viernes 11 de junio**

#### **Pautas generales:**

- La entrega es en grupos de a los sumo dos personas.
- Los algoritmos deben ser ejecutados sobre el cluster provisto por la cátedra al momento de medir los tiempos de ejecución:
  - o En IDEAS se encuentra el instructivo que explica cómo usarlo.
  - o Mediante mensajería, debe solicitar las credenciales de acceso (si aun no lo hizo).
- Se recomienda desarrollar en sus máquinas locales y usar el tiempo del cluster para las pruebas de producción.
- El tiempo de ejecución debe obtenerse sólo de la parte del algoritmo que realiza el procesamiento. Por lo tanto, NO debe incluir:
  - o Alocación y liberación de memoria
  - o Impresión en pantalla (printf)
  - o Inicialización de estructuras de datos
  - o Impresión y verificación de resultados
- El envío de los archivos debe realizarse por mensajería de IDEAS a los docentes Enzo Rucci y Adrián Pousa. Se debe enviar:
  - o Los archivos .c con el código fuente de cada ejercicio.
  - o Un informe en PDF que describa brevemente las soluciones planteadas, análisis de resultados y conclusiones.
- La entrega cuenta con un coloquio por el medio virtual establecido por la cátedra.

**Enunciado**

1. Resuelva los ejercicios 2 y 3 de la práctica 4.
2. Dada la siguiente expresión:

$$C = T + avg_{R_1} \cdot avg_{R_2} \cdot (R_1 A + R_2 B)$$

Donde  $A, B, C, T, R_1$  y  $R_2$  son matrices cuadradas de  $N \times N$ .

$avg_{R_1}$  y  $avg_{R_2}$  es el valor promedio de los elementos de la matriz  $R_1$  y  $R_2$ , respectivamente.

El elemento  $(i,j)$  de la matriz  $R_1$  debe calcularse como:

$$R_1(i,j) = (1 - T_{i,j})(1 - \cos\theta_{i,j}) + T_{i,j}\sin\theta_{i,j}$$

El elemento  $(i,j)$  de la matriz  $R_2$  debe calcularse como:

$$R_2(i,j) = (1 - T_{i,j})(1 - \sin\theta_{i,j}) + T_{i,j}\cos\theta_{i,j}$$

Donde  $T_{i,j}$  es el elemento en la posición  $(i,j)$  de la matriz  $T$ .

El ángulo  $\theta_{i,j}$ , en radianes, se obtiene de la posición  $(i,j)$  de una matriz  $M$  de  $N \times N$ . Los valores de los elementos de la matriz  $M$  están comprendidos en un rango entre 0 y  $2\pi$ .

Desarrolle 2 algoritmos que computen la expresión dada:

1. Algoritmo paralelo empleando MPI
2. Algoritmo paralelo híbrido empleando MPI+OpenMP

Los resultados deben validarse comparando la salida del algoritmo secuencial con la salida del algoritmo paralelo. Posiblemente deban tener en cuenta algún grado de error debido a la precisión en el cálculo.

Mida el tiempo de ejecución de los algoritmos en el cluster remoto. Las pruebas deben considerar la variación del tamaño de problema ( $N=\{512, 1024, 2048, 4096\}$ ) y, en el caso de los algoritmos paralelos, también la cantidad de núcleos ( $P=\{8,16,32\}$  para MPI, es decir, 1, 2 y 4 nodos, respectivamente;  $P=\{16,32\}$  para híbrido, es decir, 2 y 4 nodos, respectivamente).

Además de los algoritmos (*archivos .c*), debe elaborar un informe en PDF que describa brevemente las soluciones planteadas (puede emplear pseudo-código, diagramas, figuras, etc) e incluya las tablas con los tiempos de ejecución y valores de Speedup<sup>1</sup>, Eficiencia y overhead de comunicación para cada caso. Además, debe analizar su rendimiento y escalabilidad (individual y comparativamente):

- En el caso de  $P=8$ , compare el rendimiento del algoritmo MPI con el de Pthreads/OpenMP<sup>2</sup> de la Entrega 2.
- En el caso de  $P=\{16,32\}$ , compare el rendimiento del algoritmo MPI con el del híbrido.

Por último, recuerde aplicar las técnicas de programación y optimización vistas en clase.

---

<sup>1</sup> Como algoritmo secuencial emplee el de la Entrega 2

<sup>2</sup> El que mejor rendimiento haya tenido