

**TAREA PROGRAMADA (TP) VALOR 10%**  
**se puede trabajar en parejas**

Enviar a más tardar **Miércoles 10 de abril** antes de medianoche

## OBJETIVO

El objetivo de esta tarea es que el estudiante logre aprovechar al máximo y de manera óptima las facilidades que brinda **MPI** para hacer programación paralela en el cluster de la escuela, sin desatender la eficiencia (es decir, debe hacer un programa que se ejecute como varios procesos en paralelo el cual debería bajar el tiempo de ejecución en comparación con el tiempo que tardaría el mismo programa secuencial).

## 1. DESCRIPCIÓN DEL PROBLEMA A RESOLVER

Crear un programa en C ó C++ con comandos de MPI que pueda correr en el cluster de la escuela que resuelva lo siguiente.

- Un solo proceso (el proceso raíz o maestro) debe generar aleatoriamente dos matrices de enteros:  $A_{n \times n}$  y  $B_{n \times n}$   $n$  definido por el usuario (por facilidad se pide **que  $n$  sea múltiplo del número de procesos**), A con valores aleatorios entre **0 y 5** y B con valores aleatorios entre **0 y 2**. Una vez generadas A y B se debe:

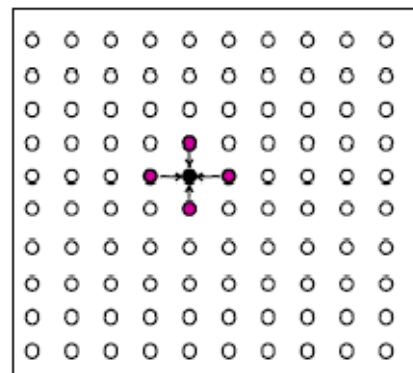
Calcular de manera distribuida entre varios procesos:

- la matriz  $M = A * B$ ,
- $tp =$  **número total de primos en M**
- el vector  $P$  tal que  $P[i] =$  **cantidad de números primos en la columna i de M**
- crear la matriz  $C$  tal que:

**Muy importante:** Los puntos **b, c y d** deben realizarse recorriendo una única vez la parte de M que le corresponde a cada proceso (note que dada una entrada de M, se sabe a qué entradas debe sumarse - o qué entradas deben sumarse a ésta- y si es primo o no)

$$C[i,j] = M[i,j] + M[i,j-1] + M[i-1,j] + M[i,j+1] + M[i+1,j]$$

Note que para las filas y columnas de los extremos, el procedimiento debe ser diferente por no tener "alrededor" todos los valores que se indican en la fórmula.



## 2. DISEÑO DE LA SOLUCIÓN PARALELIZADA:

Parte de lo que se evalúa en esta tarea es la forma en la que se **paralleliza la solución programada del problema anterior**. Es importante que lo analice con cuidado.

**RECUERDE** que para trabajar con MPI una matriz no se puede almacenar en memoria como un vector de punteros.

### 3. RESULTADOS (Desplegar en pantalla al finalizar):

- El valor de **n**,
- El número total de procesos que corrieron
- El valor de **tp** (total de valores primos de M)
- El **tiempo total** que tardó el programa desde que comenzó hasta antes de finalizar.
- El tiempo que tardó **desde que ya el usuario comunicó sus valores hasta antes de que se desplieguen resultados en pantalla y se escriban los archivos de texto**.
- Si **n**  $\leq$  100, se despliegan en pantalla también: **A, B, M, P** y **C**, de manera que se puedan distinguir fácilmente las filas y las columnas en el caso de las matrices. Si **n**  $>$  100 se copia en un archivo de texto cada uno de estos arreglos.

### 4. CÓMO TRABAJAR Y ENTREGAR SU TAREA

- Puede trabajarse en parejas
- Deje en su directorio casa del cluster su programa, el cual no puede ser modificado luego de la fecha y hora de entrega, **pero igual debe enviar el código de éste junto con la documentación**.
- Deje en su directorio casa los resultados para el caso a y el caso b (indicados más adelante), con los nombres **Aa.txt, Ba.txt, Ma.txt, Pa.txt, Ba.txt y Ab.txt, Bb.txt, Mb.txt, Pb.txt, Bb.txt**
- Documentación (la cual debe enviarse a [arqui.g1@gmail.com](mailto:arqui.g1@gmail.com) a más tardar el **miércoles 10 de abril antes de medianoche: (solo un envío por grupo de trabajo)**
  - ♦ Nombre del estudiante (s) -
  - ♦ directorio casa en el que quedaron los resultados
  - ♦ El código del programa identificado como: TP.c ó TP.cpp (en documentación interna el nombre de quienes lo programaron)
  - ♦ Un "pantallazo" con el resultado final para **caso a, caso b y caso c** descritos más abajo.
  - ♦ Una tabla con los datos descritos más abajo para los siguientes 3 casos:
    - a) **Caso a:** Corra su programa con solo un proceso, con  $n = 2000$ , (ó 200)
    - b) **Caso b:** Corra su programa con 10 procesos, con  $n = 2000$ , (ó 200)
    - c) **Caso c:** Corra su programa con 10 procesos, con  $n = 30$ ,
    - d) Con los valores que se despliegan al final de cada corrida complete esta tabla y agréguela a su documentación:

Resultados	Caso a 1 proceso, $n = 2000$	Caso b 10 procesos, $n = 2000$	Caso c 10 procesos, $n = 30$
Tiempo total			
Tiempo sin despliegues en pantalla ni escrituras de archivos			

- ♦ Un pequeño análisis (**análisis, no descripción**) de los resultados anteriores (media página máximo)
- ♦ Si no se logró que corriera correctamente el programa, explicaciones del por qué e idea de cómo resolverlo