

# Ejercicio/Práctica de MPI

Computación de Altas Prestaciones

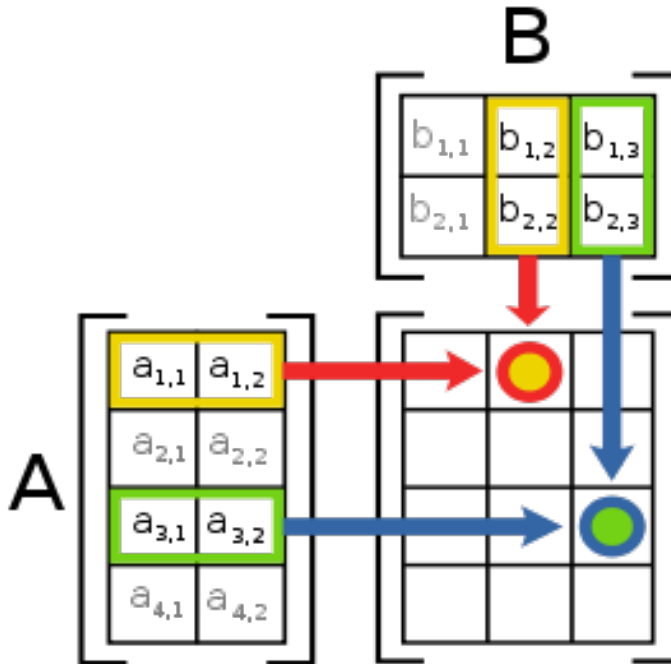
Curso: 2015/2016

## 1 Descripción del Problema

Se desea realizar un programa que realice una multiplicación de matrices usando MPI. Es decir, dada dos matrices  $A$  y  $B$  el programa deberá de realizar la multiplicación  $A \times B$ :

$$A \times B(i, j) = \sum_{k=1}^N A(i, k) * B(k, i)$$

O visualmente:



### 1.1 Facilidades

Se ofrece una versión secuencial del programa, `sec_multmatrixseq.c`, junto con un fichero `CMakeLists.txt`, por comodidad. El método multiplicación no debe ser modificado, simplemente hay que lla-

marlo con las submatrices correspondientes. Se pueden hacer las compilaciones de forma manual, o usar el programa [CMake](#) que permite compilar fácilmente proyectos.

### 1. Compilación

Para compilar se puede realizar:

```
cmake . # Genera el programa Makefile
make # Para compilar
```

(por defecto ya compila para poder repurar).

### 2. Depuración

La depuración puede ser compleja por varios factores:

- Si algún parámetro en algunas de las funciones `MPI_...` no es correcto, el programa corta de forma abrupta.
- Los mensajes por pantalla se muestran desordenados, incluso desactivando *buffers* o poniendo barreras.

La forma más sencilla de depurar es usando el clásico gdb. Se puede ejecutar en gdb en varias ventanas, cada una con un programa, con la siguiente sintaxis:

```
mpirun -n 4 xterm -e gdb ./mpi_multmatrix
```

De esta manera se abre una ventana para cada proceso, y se puede ir viendo si cada proceso recibe o envía lo que debe.

## 1.2 Requisitos

Para poder probar los conocimientos avanzados, y darle un poco de *gracia* a la práctica, no se aceptará una versión que envíe los datos múltiples veces con un **Send** ó **ISend**, si no que será necesario el uso del **Scatterv** para repartir cada matriz de una vez (una vez para la matriz 1 y otra para la matriz 2, claro). Y luego se deberá recoger el resultado en el proceso inicial (rank=0) que mostrará el resultado.

## 1.3 Versión simplificada

En un primer intento, se recomienda probar con una versión simplificada, usando matrices cuadradas, o algún tipo de limitación.

## 1.4 Versión completa

La versión completa requerirá ser genérica. Eso implica:

- Verificar que el número de procesos sea adecuado.

- Repartir la primera matrix por filas y la segunda por columnas, para que cada proceso tenga los datos necesarios.
- No habrá limitación sobre el número de filas o columnas de las matrices.
- No obstante, Se podrá suponer que el resultado tendrá menores elementos que las matrices. Esto implica que, por ejemplo, podremos hacer  $(2 \times 4) \times (4 \times 2) \Rightarrow 2 \times 2$ , y no  $(4 \times 2) \times (2 \times 4) \Rightarrow 4 \times 4$ , ya que ese segundo caso es bastante más complejo. De todas formas, se valorará lo generalizable que sea.