

Programación con MPI / Programación híbrida

<https://prod.liveshare.vsengsaas.visualstudio.com/join?6633641252D11F6BBA48E78A4EC D6B8C0603>

Ejercicios

1. Revisar el código mpi-simple.c. Compile y ejecute el código. Modifíquelo para que los procesos se comuniquen en forma de anillo: el proceso i debe enviarle un mensaje al proceso i+1, a excepción del último que debe comunicarse con el 0.

comunicarse

2. Los códigos blocking.c y non-blocking.c siguen el patrón master-worker, donde los procesos worker le envían un mensaje de texto al master empleando operaciones de comunicación bloqueantes y no bloqueantes, respectivamente.

- Compile y ejecute ambos códigos usando $P=\{4,8,16\}$ (no importa que el número de núcleos sea menor que la cantidad de procesos). ¿Cuál de los dos retorna antes el control?

Nota: para poder obtener correctamente el tiempo total tuvimos que cambiar en ambos archivos la línea:

```
resolution = MPI_Wtick();
```

por

```
resolution = toc - tic;
```

Resultados:

Con **P=4**

Bloqueante:

```
Tiempo transcurrido 0.000001 (s): proceso 0, llamando a MPI_Recv() [bloqueante] (fuente rank 1)
Tiempo transcurrido 2.000218 (s): proceso 0, MPI_Recv() devolvio control con mensaje: Hola Mundo! Soy el proceso 1
Tiempo transcurrido 2.000231 (s): proceso 0, llamando a MPI_Recv() [bloqueante] (fuente rank 2)
Tiempo transcurrido 4.000190 (s): proceso 0, MPI_Recv() devolvio control con mensaje: Hola Mundo! Soy el proceso 2
Tiempo transcurrido 4.000202 (s): proceso 0, llamando a MPI_Recv() [bloqueante] (fuente rank 3)
Tiempo transcurrido 6.000226 (s): proceso 0, MPI_Recv() devolvio control con mensaje: Hola Mundo! Soy el proceso 3

Tiempo total = 6.000226 (s)
```

No bloqueante:

```
Tiempo transcurrido 0.000001 (s): proceso 0, llamando a MPI_IRecv() [no bloqueante] (fuente rank 1)
Tiempo transcurrido 0.000047 (s): proceso 0, MPI_IRecv() devolvio el control..
    ..pero el mensaje no fue aun recibido..
Tiempo transcurrido 2.000321 (s): proceso 0, operacion receive completa con mensaje: Hola Mundo! Soy el proceso 1
Tiempo transcurrido 2.000334 (s): proceso 0, llamando a MPI_IRecv() [no bloqueante] (fuente rank 2)
Tiempo transcurrido 2.000346 (s): proceso 0, MPI_IRecv() devolvio el control..
    ..pero el mensaje no fue aun recibido..
Tiempo transcurrido 4.000249 (s): proceso 0, operacion receive completa con mensaje: Hola Mundo! Soy el proceso 2
Tiempo transcurrido 4.000284 (s): proceso 0, llamando a MPI_IRecv() [no bloqueante] (fuente rank 3)
Tiempo transcurrido 4.000298 (s): proceso 0, MPI_IRecv() devolvio el control..
    ..pero el mensaje no fue aun recibido..
Tiempo transcurrido 6.000062 (s): proceso 0, operacion receive completa con mensaje: Hola Mundo! Soy el proceso 3

Tiempo total = 6.000062 (s)
```

Con P=8

Bloqueante:

```
Tiempo transcurrido 0.000002 (s): proceso 0, llamando a MPI_Recv() [bloqueante] (fuente rank 1)
Tiempo transcurrido 2.000125 (s): proceso 0, MPI_Recv() devolvio control con mensaje: Hola Mundo! Soy el proceso 1
Tiempo transcurrido 2.000138 (s): proceso 0, llamando a MPI_Recv() [bloqueante] (fuente rank 2)
Tiempo transcurrido 4.000125 (s): proceso 0, MPI_Recv() devolvio control con mensaje: Hola Mundo! Soy el proceso 2
Tiempo transcurrido 4.000137 (s): proceso 0, llamando a MPI_Recv() [bloqueante] (fuente rank 3)
Tiempo transcurrido 6.000197 (s): proceso 0, MPI_Recv() devolvio control con mensaje: Hola Mundo! Soy el proceso 3
Tiempo transcurrido 6.000209 (s): proceso 0, llamando a MPI_Recv() [bloqueante] (fuente rank 4)
Tiempo transcurrido 8.000146 (s): proceso 0, MPI_Recv() devolvio control con mensaje: Hola Mundo! Soy el proceso 4
Tiempo transcurrido 8.000160 (s): proceso 0, llamando a MPI_Recv() [bloqueante] (fuente rank 5)
Tiempo transcurrido 10.000064 (s): proceso 0, MPI_Recv() devolvio control con mensaje: Hola Mundo! Soy el proceso 5
Tiempo transcurrido 10.000080 (s): proceso 0, llamando a MPI_Recv() [bloqueante] (fuente rank 6)
Tiempo transcurrido 12.000133 (s): proceso 0, MPI_Recv() devolvio control con mensaje: Hola Mundo! Soy el proceso 6
Tiempo transcurrido 12.000146 (s): proceso 0, llamando a MPI_Recv() [bloqueante] (fuente rank 7)
Tiempo transcurrido 14.000142 (s): proceso 0, MPI_Recv() devolvio control con mensaje: Hola Mundo! Soy el proceso 7

Tiempo total = 14.000142 (s)
```

No bloqueante:

```
Tiempo transcurrido 0.000002 (s): proceso 0, llamando a MPI_IRecv() [no bloqueante] (fuente rank 1)
Tiempo transcurrido 0.000046 (s): proceso 0, MPI_IRecv() devolvio el control..
    ..pero el mensaje no fue aun recibido..
Tiempo transcurrido 2.000123 (s): proceso 0, operacion receive completa con mensaje: Hola Mundo! Soy el proceso 1
Tiempo transcurrido 2.000135 (s): proceso 0, llamando a MPI_IRecv() [no bloqueante] (fuente rank 2)
Tiempo transcurrido 2.000143 (s): proceso 0, MPI_IRecv() devolvio el control..
    ..pero el mensaje no fue aun recibido..
Tiempo transcurrido 4.000271 (s): proceso 0, operacion receive completa con mensaje: Hola Mundo! Soy el proceso 2
Tiempo transcurrido 4.000284 (s): proceso 0, llamando a MPI_IRecv() [no bloqueante] (fuente rank 3)
Tiempo transcurrido 4.000292 (s): proceso 0, MPI_IRecv() devolvio el control..
    ..pero el mensaje no fue aun recibido..
Tiempo transcurrido 6.000201 (s): proceso 0, operacion receive completa con mensaje: Hola Mundo! Soy el proceso 3
Tiempo transcurrido 6.000213 (s): proceso 0, llamando a MPI_IRecv() [no bloqueante] (fuente rank 4)
Tiempo transcurrido 6.000221 (s): proceso 0, MPI_IRecv() devolvio el control..
    ..pero el mensaje no fue aun recibido..
Tiempo transcurrido 8.000088 (s): proceso 0, operacion receive completa con mensaje: Hola Mundo! Soy el proceso 4
Tiempo transcurrido 8.000100 (s): proceso 0, llamando a MPI_IRecv() [no bloqueante] (fuente rank 5)
Tiempo transcurrido 8.000108 (s): proceso 0, MPI_IRecv() devolvio el control..
    ..pero el mensaje no fue aun recibido..
Tiempo transcurrido 10.000160 (s): proceso 0, operacion receive completa con mensaje: Hola Mundo! Soy el proceso 5
Tiempo transcurrido 10.000175 (s): proceso 0, llamando a MPI_IRecv() [no bloqueante] (fuente rank 6)
Tiempo transcurrido 10.000184 (s): proceso 0, MPI_IRecv() devolvio el control..
    ..pero el mensaje no fue aun recibido..
Tiempo transcurrido 12.000217 (s): proceso 0, operacion receive completa con mensaje: Hola Mundo! Soy el proceso 6
Tiempo transcurrido 12.000231 (s): proceso 0, llamando a MPI_IRecv() [no bloqueante] (fuente rank 7)
Tiempo transcurrido 12.000239 (s): proceso 0, MPI_IRecv() devolvio el control..
    ..pero el mensaje no fue aun recibido..
Tiempo transcurrido 13.999997 (s): proceso 0, operacion receive completa con mensaje: Hola Mundo! Soy el proceso 7

Tiempo total = 13.999997 (s)
```

P=16

Bloqueante:

```
Tiempo transcurrido 0.000002 (s): proceso 0, llamando a MPI_Recv() [bloqueante] (fuente rank 1)
Tiempo transcurrido 2.000249 (s): proceso 0, MPI_Recv() devolvio control con mensaje: Hola Mundo! Soy el proceso 1
Tiempo transcurrido 2.000263 (s): proceso 0, llamando a MPI_Recv() [bloqueante] (fuente rank 2)
Tiempo transcurrido 4.000143 (s): proceso 0, MPI_Recv() devolvio control con mensaje: Hola Mundo! Soy el proceso 2
Tiempo transcurrido 4.000155 (s): proceso 0, llamando a MPI_Recv() [bloqueante] (fuente rank 3)
Tiempo transcurrido 6.000225 (s): proceso 0, MPI_Recv() devolvio control con mensaje: Hola Mundo! Soy el proceso 3
Tiempo transcurrido 6.000237 (s): proceso 0, llamando a MPI_Recv() [bloqueante] (fuente rank 4)
Tiempo transcurrido 8.000455 (s): proceso 0, MPI_Recv() devolvio control con mensaje: Hola Mundo! Soy el proceso 4
Tiempo transcurrido 8.000468 (s): proceso 0, llamando a MPI_Recv() [bloqueante] (fuente rank 5)
Tiempo transcurrido 10.000235 (s): proceso 0, MPI_Recv() devolvio control con mensaje: Hola Mundo! Soy el proceso 5
Tiempo transcurrido 10.000248 (s): proceso 0, llamando a MPI_Recv() [bloqueante] (fuente rank 6)
Tiempo transcurrido 12.000230 (s): proceso 0, MPI_Recv() devolvio control con mensaje: Hola Mundo! Soy el proceso 6
Tiempo transcurrido 12.000244 (s): proceso 0, llamando a MPI_Recv() [bloqueante] (fuente rank 7)
Tiempo transcurrido 14.000592 (s): proceso 0, MPI_Recv() devolvio control con mensaje: Hola Mundo! Soy el proceso 7
Tiempo transcurrido 14.000605 (s): proceso 0, llamando a MPI_Recv() [bloqueante] (fuente rank 8)
Tiempo transcurrido 16.000531 (s): proceso 0, MPI_Recv() devolvio control con mensaje: Hola Mundo! Soy el proceso 8
Tiempo transcurrido 16.000546 (s): proceso 0, llamando a MPI_Recv() [bloqueante] (fuente rank 9)
Tiempo transcurrido 18.002267 (s): proceso 0, MPI_Recv() devolvio control con mensaje: Hola Mundo! Soy el proceso 9
Tiempo transcurrido 18.002280 (s): proceso 0, llamando a MPI_Recv() [bloqueante] (fuente rank 10)
Tiempo transcurrido 20.000350 (s): proceso 0, MPI_Recv() devolvio control con mensaje: Hola Mundo! Soy el proceso 10
Tiempo transcurrido 20.000362 (s): proceso 0, llamando a MPI_Recv() [bloqueante] (fuente rank 11)
Tiempo transcurrido 22.000494 (s): proceso 0, MPI_Recv() devolvio control con mensaje: Hola Mundo! Soy el proceso 11
Tiempo transcurrido 22.000506 (s): proceso 0, llamando a MPI_Recv() [bloqueante] (fuente rank 12)
Tiempo transcurrido 24.000647 (s): proceso 0, MPI_Recv() devolvio control con mensaje: Hola Mundo! Soy el proceso 12
Tiempo transcurrido 24.000659 (s): proceso 0, llamando a MPI_Recv() [bloqueante] (fuente rank 13)
Tiempo transcurrido 26.000225 (s): proceso 0, MPI_Recv() devolvio control con mensaje: Hola Mundo! Soy el proceso 13
Tiempo transcurrido 26.000240 (s): proceso 0, llamando a MPI_Recv() [bloqueante] (fuente rank 14)
Tiempo transcurrido 28.000584 (s): proceso 0, MPI_Recv() devolvio control con mensaje: Hola Mundo! Soy el proceso 14
Tiempo transcurrido 28.000597 (s): proceso 0, llamando a MPI_Recv() [bloqueante] (fuente rank 15)
Tiempo transcurrido 30.006011 (s): proceso 0, MPI_Recv() devolvio control con mensaje: Hola Mundo! Soy el proceso 15

Tiempo total = 30.006011 (s)
```

No bloqueante:

```
Tiempo transcurrido 0.000001 (s): proceso 0, llamando a MPI_IRecv() [no bloqueante] (fuente rank 1)
Tiempo transcurrido 0.000088 (s): proceso 0, MPI_IRecv() devolvio el control..
..pero el mensaje no fue aun recibido..
Tiempo transcurrido 2.000229 (s): proceso 0, operacion receive completa con mensaje: Hola Mundo! Soy el proceso 1
Tiempo transcurrido 2.000233 (s): proceso 0, llamando a MPI_IRecv() [no bloqueante] (fuente rank 2)
Tiempo transcurrido 2.000242 (s): proceso 0, MPI_IRecv() devolvio el control..
..pero el mensaje no fue aun recibido..
Tiempo transcurrido 4.000010 (s): proceso 0, operacion receive completa con mensaje: Hola Mundo! Soy el proceso 2
Tiempo transcurrido 4.000022 (s): proceso 0, llamando a MPI_IRecv() [no bloqueante] (fuente rank 3)
Tiempo transcurrido 4.000029 (s): proceso 0, MPI_IRecv() devolvio el control..
..pero el mensaje no fue aun recibido..
Tiempo transcurrido 6.000118 (s): proceso 0, operacion receive completa con mensaje: Hola Mundo! Soy el proceso 3
Tiempo transcurrido 6.000131 (s): proceso 0, llamando a MPI_IRecv() [no bloqueante] (fuente rank 4)
Tiempo transcurrido 6.000138 (s): proceso 0, MPI_IRecv() devolvio el control..
..pero el mensaje no fue aun recibido..
Tiempo transcurrido 8.000125 (s): proceso 0, operacion receive completa con mensaje: Hola Mundo! Soy el proceso 4
Tiempo transcurrido 8.000138 (s): proceso 0, llamando a MPI_IRecv() [no bloqueante] (fuente rank 5)
Tiempo transcurrido 8.000145 (s): proceso 0, MPI_IRecv() devolvio el control..
..pero el mensaje no fue aun recibido..
Tiempo transcurrido 10.000104 (s): proceso 0, operacion receive completa con mensaje: Hola Mundo! Soy el proceso 5
Tiempo transcurrido 10.000119 (s): proceso 0, llamando a MPI_IRecv() [no bloqueante] (fuente rank 6)
Tiempo transcurrido 10.000127 (s): proceso 0, MPI_IRecv() devolvio el control..
..pero el mensaje no fue aun recibido..
Tiempo transcurrido 11.999951 (s): proceso 0, operacion receive completa con mensaje: Hola Mundo! Soy el proceso 6
Tiempo transcurrido 11.999964 (s): proceso 0, llamando a MPI_IRecv() [no bloqueante] (fuente rank 7)
Tiempo transcurrido 11.999972 (s): proceso 0, MPI_IRecv() devolvio el control..
..pero el mensaje no fue aun recibido..
Tiempo transcurrido 14.000628 (s): proceso 0, operacion receive completa con mensaje: Hola Mundo! Soy el proceso 7
Tiempo transcurrido 14.000641 (s): proceso 0, llamando a MPI_IRecv() [no bloqueante] (fuente rank 8)
Tiempo transcurrido 14.000650 (s): proceso 0, MPI_IRecv() devolvio el control..
..pero el mensaje no fue aun recibido..
Tiempo transcurrido 15.999934 (s): proceso 0, operacion receive completa con mensaje: Hola Mundo! Soy el proceso 8
Tiempo transcurrido 15.999948 (s): proceso 0, llamando a MPI_IRecv() [no bloqueante] (fuente rank 9)
Tiempo transcurrido 15.999956 (s): proceso 0, MPI_IRecv() devolvio el control..
..pero el mensaje no fue aun recibido..
Tiempo transcurrido 18.000112 (s): proceso 0, operacion receive completa con mensaje: Hola Mundo! Soy el proceso 9
Tiempo transcurrido 18.000124 (s): proceso 0, llamando a MPI_IRecv() [no bloqueante] (fuente rank 10)
Tiempo transcurrido 18.000134 (s): proceso 0, MPI_IRecv() devolvio el control..
..pero el mensaje no fue aun recibido..
Tiempo transcurrido 20.000104 (s): proceso 0, operacion receive completa con mensaje: Hola Mundo! Soy el proceso 10
Tiempo transcurrido 20.000197 (s): proceso 0, llamando a MPI_IRecv() [no bloqueante] (fuente rank 11)
Tiempo transcurrido 20.000205 (s): proceso 0, MPI_IRecv() devolvio el control..
..pero el mensaje no fue aun recibido..
Tiempo transcurrido 22.001488 (s): proceso 0, operacion receive completa con mensaje: Hola Mundo! Soy el proceso 11
Tiempo transcurrido 22.001500 (s): proceso 0, llamando a MPI_IRecv() [no bloqueante] (fuente rank 12)
Tiempo transcurrido 22.001508 (s): proceso 0, MPI_IRecv() devolvio el control..
..pero el mensaje no fue aun recibido..
Tiempo transcurrido 24.000323 (s): proceso 0, operacion receive completa con mensaje: Hola Mundo! Soy el proceso 12
Tiempo transcurrido 24.000337 (s): proceso 0, llamando a MPI_IRecv() [no bloqueante] (fuente rank 13)
Tiempo transcurrido 24.000346 (s): proceso 0, MPI_IRecv() devolvio el control..
..pero el mensaje no fue aun recibido..
Tiempo transcurrido 26.000166 (s): proceso 0, operacion receive completa con mensaje: Hola Mundo! Soy el proceso 13
Tiempo transcurrido 26.000178 (s): proceso 0, llamando a MPI_IRecv() [no bloqueante] (fuente rank 14)
Tiempo transcurrido 26.000186 (s): proceso 0, MPI_IRecv() devolvio el control..
..pero el mensaje no fue aun recibido..
Tiempo transcurrido 28.001570 (s): proceso 0, operacion receive completa con mensaje: Hola Mundo! Soy el proceso 14
Tiempo transcurrido 28.001646 (s): proceso 0, llamando a MPI_IRecv() [no bloqueante] (fuente rank 15)
Tiempo transcurrido 28.001660 (s): proceso 0, MPI_IRecv() devolvio el control..
..pero el mensaje no fue aun recibido..
Tiempo transcurrido 30.000237 (s): proceso 0, operacion receive completa con mensaje: Hola Mundo! Soy el proceso 15

Tiempo total = 30.000237 (s)
```

Conclusiones: observando los resultados obtenidos podemos concluir en que el algoritmo no bloqueante devuelve mucho antes el control ya que no espera a que el proceso receptor lo reciba. No obstante, usar `MPI_IRecv()` no asegura que la operación haya terminado. Para asegurarse de no continuar cuando el mensaje aún no se recibe, en este algoritmo se debe usar el `MPI_Wait()`, operación que duerme al proceso receptor hasta que la operación indicada haya finalizado.

En el caso de `blocking.c`, se utiliza el receive bloqueante `MPI_Recv()`, que retorna el control recién cuando se copia el mensaje en el buffer del receptor, lo que hace que se tarde más en devolver el control.

Usar una estrategia no bloqueante puede ser útil cuando la tarea que envía el mensaje tiene algo para hacer mientras se termina de hacer la comunicación.

Ejecutamos varias veces ambos programas y los resultados fueron siempre bastante similares entre sí.

- En el caso de la versión no bloqueante, ¿qué sucede si se elimina la operación `MPI_Wait()` (línea 52)? ¿Se imprimen correctamente los mensajes enviados? ¿Por qué?

```
Tiempo transcurrido 0.000001 (s): proceso 0, llamando a MPI_IRecv() [no bloqueante] (fuente rank 1)
Tiempo transcurrido 0.000042 (s): proceso 0, MPI_IRecv() devolvio el control..
    ..pero el mensaje no fue aun recibido..
Tiempo transcurrido 0.000064 (s): proceso 0, operacion receive completa con mensaje: No deberia estar leyendo esta frase.
Tiempo transcurrido 0.000079 (s): proceso 0, llamando a MPI_IRecv() [no bloqueante] (fuente rank 2)
Tiempo transcurrido 0.000092 (s): proceso 0, MPI_IRecv() devolvio el control..
    ..pero el mensaje no fue aun recibido..
Tiempo transcurrido 0.000105 (s): proceso 0, operacion receive completa con mensaje: No deberia estar leyendo esta frase.
Tiempo transcurrido 0.000112 (s): proceso 0, llamando a MPI_IRecv() [no bloqueante] (fuente rank 3)
Tiempo transcurrido 0.000121 (s): proceso 0, MPI_IRecv() devolvio el control..
    ..pero el mensaje no fue aun recibido..
Tiempo transcurrido 0.000134 (s): proceso 0, operacion receive completa con mensaje: No deberia estar leyendo esta frase.

Tiempo total = 0.000134 (s)
```

Al eliminar la operación `MPI_Wait()` la recepción de mensajes empieza a ser insegura ya que cada emisor tarda ~2 segundos en enviar un mensaje y al no esperar que la comunicación se termine de forma correcta el mensaje que se imprime será el que está almacenado en el buffer como default. En lugar de imprimirse los mensajes “Hola, Mundo! Soy el proceso x” se imprime “No debería estar leyendo esta frase”, que es un indicador de que el master continuó ejecutándose aunque aún no tenía el mensaje disponible. La ejecución del programa tarda mucho menos ya que el Maestro continúa con su ejecución sin esperar a sincronizar ningún mensaje.

3. Los códigos `blocking-ring.c` y `non-blocking-ring.c` comunican a los procesos en forma de anillo empleando operaciones bloqueantes y no bloqueantes, respectivamente. Compile y ejecute ambos códigos empleando $P=\{4,8,16\}$ (no importa que el número de núcleos sea

menor que la cantidad de procesos) y $N=\{10000000, 20000000, 40000000, \dots\}$. ¿Cuál de los dos algoritmos requiere menos tiempo de comunicación? ¿Por qué?

Nota: Para el caso de $P=16$, agregue la línea `--overcommit` al script de de SLURM y el flag `--oversubscribe` al comando `mpirun`.

Nota: si ejecutamos con 400000 nos da el siguiente error:

```
Primary job terminated normally, but 1 process returned
a non-zero exit code. Per user-direction, the job has been aborted.

mpirun noticed that process rank 0 with PID 31048 on node nodo1 exited on signal 9 (Killed).
```

Por lo que realizamos las pruebas con $N=\{10000000, 20000000, 30000000\}$

Resultados de non-blocking-ring:

Tamaño del vector	Nº de procesos	Tiempo total comunicación (s)	Promedio de cada comunicación (s)
10,000,000	4	0.132483	0.000045
10,000,000	8	0.303695	0.037014
10,000,000	16	0.604290	0.222973
20,000,000	4	0.265010	0.066788
20,000,000	8	0.588422	0.084487
20,000,000	16	1.256911	0.386803
30,000,000	4	0.398245	0.131843
30,000,000	8	0.854631	0.149723
30,000,000	16	1.899944	0.529197

Resultados de blocking-ring:

Tamaño del vector	Nº de procesos	Tiempo total comunicación (s)	Promedio de cada comunicación (s)
10,000,000	4	0.252692	0.063173
10,000,000	8	0.571336	0.071417
10,000,000	16	1.804412	0.112776
20,000,000	4	0.503000	0.125750
20,000,000	8	1.136666	0.142083
20,000,000	16	3.700002	0.231250
30,000,000	4	0.774517	0.193629
30,000,000	8	1.713061	0.214133
30,000,000	16	5.643920	0.352745

Conclusiones:

. El algoritmo `mpi_matmul.c` computa una multiplicación de matrices cuadradas empleando comunicaciones punto a punto:

- Compile y ejecute el código empleando $N=\{512,1024,2048\}$ usando todos los núcleos de 1 y 2 nodos.

comunicarse

- Revise las secciones de código donde se realiza la comunicación de las matrices. Analice el patrón de comunicación y piense si es posible emplear comunicaciones colectivas en lugar de a punto a punto. En ese caso, modifique el código original, compile y ejecute la nueva versión. ¿Se mejora la legibilidad? ¿Se logra mejorar el rendimiento? ¿Por qué?

comunicarse

5. Desarrolle un algoritmo paralelo que resuelva la expresión $R = AB + CD + EF$, donde A, B, C, D, E y F son matrices cuadradas de $N \times N$. Ejecute para $N = \{512, 1024, 2048\}$ con $P=\{2,4,8,16\}$.

comunicarse

6. Desarrolle un algoritmo paralelo que dado un vector V de tamaño N obtenga el valor máximo, el valor mínimo y valor promedio de sus elementos. Ejecute para $P=\{2,4,8,16\}$ variando el valor de N.

comunicarse

7. Desarrolle una versión híbrida (MPI+OpenMP) de la multiplicación de matrices. Replique el análisis realizado para el algoritmo puro MPI (ejercicio 4) y compare sus rendimientos. ¿Cuál es mejor? ¿Por qué?

comunicarse