

**Practica 1 Computación de Altas Prestaciones
MPI**

**Autores: Andrés Tena De Tena y
Gabriel Morales Dato
Titulación: Ingeniería de Computadores
Curso: 2023/2024**

Fecha: 1 de abril de 2024

Índice

1. Resultados de la práctica	2
1.1. Versión sin utilizar MPI	2
1.2. Envío de la linea descifrada desde los esclavos	2
1.3. Envío de la clave desde los esclavos	3
2. Análisis de los resultados	4

1. Resultados de la práctica

A continuación se exponen los resultados obtenidos en las diferentes versiones de la práctica:

1.1. Versión sin utilizar MPI

En esta versión solo se ejecuta un hilo que realiza todo el trabajo de forma secuencial.

Tiempos de ejecución:

- 23 líneas x 138 caracteres x 5 rotores: 1122.00 ms
- 8 líneas x 93 caracteres x 5 rotores: 495.00 ms
- 9 líneas x 33 caracteres x 2 rotores: 0.22 ms

1.2. Envío de la línea descifrada desde los esclavos

1. Envío de una sola línea por esclavo:

- Descripción: El maestro envía las líneas a los esclavos de una en una y las recibe descifradas para luego mostrarlas.
- Tiempos de ejecución:
 - 23 líneas x 138 caracteres x 5 rotores: 293.00 ms (mejora un 74 %)
 - 8 líneas x 93 caracteres x 5 rotores: 132.00 ms (mejora un 73 %)
 - 9 líneas x 33 caracteres x 2 rotores: 0.40 ms (empeora un 82 %)

2. Envío de varias líneas por esclavo:

- Descripción: El maestro envía las líneas a los esclavos por bloques y las recibe descifradas para luego mostrarlas.
- Tiempos de ejecución sin utilizar MPI:
 - 23 líneas x 138 caracteres x 5 rotores: 2263.82 ms
 - 8 líneas x 93 caracteres x 5 rotores: 783.10 ms
 - 9 líneas x 33 caracteres x 2 rotores: 4.00 ms
- Tiempos de ejecución:
 - 23 líneas x 138 caracteres x 5 rotores: 1307.38 ms (mejora un 42 %)
 - 8 líneas x 93 caracteres x 5 rotores: 562.95 ms (mejora un 28 %)
 - 9 líneas x 33 caracteres x 2 rotores: 7.10 ms (empeora un 76 %)

1.3. Envío de la clave desde los esclavos

1. Envío de una linea por esclavo:

- Descripción: El maestro envía las lineas a los esclavos de una en una y los esclavos devuelven únicamente la clave para descifrarlas. El maestro descifra y muestra la linea correspondiente con la clave recibida.
- Tiempos de ejecución:
 - 23 lineas x 138 caracteres x 5 rotores: 304.00 ms (mejora un 73 %)
 - 8 lineas x 93 caracteres x 5 rotores: 135.00 ms (mejora un 73 %)
 - 9 lineas x 33 caracteres x 2 rotores: 0.50 ms (empeora un 127 %)

Observaciones:

- Todos los programas los hemos compilado con -O2 para partir del mismo nivel de optimización.
- Todas las mejoras se han calculado respecto a la versión sin utilizar MPI.
- Con el mensaje de 51 lineas x 155 caracteres no se han podido llegar a obtener conclusiones, ya que todos los códigos tardaban tanto que la consola acababa matando el proceso.

2. Análisis de los resultados

- **¿Hay mejora considerable en cuanto al tiempo de ejecución usando MPI o sin usarlo? ¿Por qué?**

En cuanto en el primer mensaje cifrado, el tiempo empleado por los códigos que utilizan MPI es incluso mayor que el empleado por el código básico, que no utiliza MPI. Pero cuanto más se amplía dicho mensaje cifrado el tiempo utilizado por el código sin MPI aumenta mucho más rápido y es mayor que los códigos que utilizan MPI. En el segundo mensaje el código MPI por bloques es más lento que el utilizado que no utiliza bloques. Aunque en el tercer mensaje el código que emplea bloques obtiene mejores resultados que el que no utiliza bloques. Finalmente, con el cuarto mensaje no se han podido llegar a obtener conclusiones, ya que todos los códigos tardaban tanto que la consola acababa matando el proceso. Por tanto, si hay una mejora considerable cuando el mensaje a descifrar posee un tamaño grande, esto es debido a que esta acción de descifrado es altamente paralelizable y el descifrar una línea no influye para nada en descifrar la siguiente línea.

- **¿Tiene sentido utilizar un esquema maestro/esclavo? ¿Por qué?**

No sería necesario usar este esquema, ya que podríamos mostrar el resultado cuando el último nodo termine de descifrar. Eso sí, para ello sería necesario que todos los nodos tuviesen memoria compartida, de la matriz donde se va almacenando el mensaje descifrado.

- **¿Cómo se utiliza la comunicación entre procesos en la solución?**

En esta solución se ha utilizado el esquema maestro/esclavo, en el cual el maestro es el encargado de enviar si hay trabajo para los nodos y de recibir las líneas descifradas, procedentes de los distintos esclavos, para posteriormente juntarlas en orden y mostrarlas.