

Práctica Nro. 4

Programación con MPI

Se seleccionará un ejercicio que los alumnos deberán entregar en grupos de dos personas.

Pautas:

Compilar en Linux OpenMPI:

`mpicc -o salidaEjecutable archivoFuente`

Ejecutar en OpenMPI:

En una sola máquina:

`mpirun -np cantidadDeProcesos ejecutable`

En un cluster de máquinas:

`mpirun -np cantidadDeProcesos -machinefile archivoMaquinas ejecutable`

El formato de archivo de máquinas es:

`maquina1 slots=cantidad de procesadores de la maquina1`

`maquina2 slots=cantidad de procesadores de la maquina2`

...

`maquinaN slots=cantidad de procesadores de la maquinaN`

En todos los ejercicios implementar el algoritmo secuencial y el algoritmo paralelo.

Probar el algoritmo paralelo sobre:

- 1) Una máquina con 4 procesos*
- 2) Dos máquinas con 2 procesos por máquina (4 procesos)*
- 3) Dos máquinas con 8 procesos.*

Realizar el análisis de escalabilidad y analizar el overhead introducido por comunicación.

1. Resolver una multiplicación de matrices de $N \times N$ y analizar los tiempos de comunicación utilizando:

- a) Operaciones punto a punto.
- b) Operaciones colectivas

Utilizar tamaños de matrices de 512, 1024 y 2048.

2. Realizar un algoritmo paralelo que dada una matriz A de $N \times N$ obtenga el valor máximo, el valor mínimo y valor promedio de A , luego debe armar una matriz B de la siguiente forma:

- Si el elemento $a_{i,j} < \text{promedio}(A)$ entonces $b_{i,j} = \min(A)$.
- Si el elemento $a_{i,j} > \text{promedio}(A)$ entonces $b_{i,j} = \max(A)$.
- Si el elemento $a_{i,j} = \text{promedio}(A)$ entonces $b_{i,j} = \text{promedio}(A)$.

3. Dado un texto representado por un vector T de tamaño N , se debe realizar un algoritmo paralelo que obtenga la lista de palabras de T , y luego determine la cantidad de veces que aparece cada palabra en el texto quedándose sólo con las cinco palabras más frecuentes.

4. Realizar un algoritmo paralelo que ordene un vector de N elementos por mezcla.