

Configuración cluster MPI

Iñigo Manuel Diez Canseco Fuentes
Univesidad Católica San Pablo
Arequipa, Perú
Email: inigo.diezcanseco@ucsp.edu.pe

I. INTRODUCCIÓN DEL TRABAJO

El trabajo a realizar se trata de un clúster MPI el cual debe juntarse con el algoritmo *odd-even sort* y hacer que funcione de manera paralela y distribuida.

- Subir su informe de todo lo realizado en la configuración del cluster en MPI.
- Probar su cluster con el algoritmo odd-even sort paralelo.[1]

Los códigos de prueba y del cuadro que se mostrara más adelante del documento se encuentran en el siguiente GITHUB.¹

II. CLUSTER MPI

Este algoritmo es similar al *bubble sort* tradicional, la diferencia es que en lugar de comparar los elementos contiguos en la lista, se comparan los elementos con índices pares/impares en momentos distintos. Con esta idea se utiliza un array de bools para poder verificar si es que con el primer *for* se ordenó todo y si no pasar al siguiente.

Se trató de utilizar las funciones MPI para hacer que la función Odd-even pudiera enviarse mensajes a través de cada proceso y separar los *threads* para mejorar su eficiencia, la cual se vería reflejada en el cuadro a continuación.

III. CUADRO COMPARATIVO

A continuación se presentará un cuadro comparativo del algoritmo de Odd-Even sort con sus diferentes variaciones en tiempo microsegundos(us):

Cluster MPI				
	Tamaño del Array (microsegundos us)			
Algoritmos	100	1000	10000	100000
Odd even Normal	18	1473	1692261	17838391
Parcial MPI	37	3555	338621	15596659
Experimento	x	x	x	x

Figura 1. Uniform Memory Access, Referencia de las diapositivas.

REFERENCIAS

- [1] P. Pacheco, *An introduction to parallel programming*. Elsevier, 2011.

¹https://github.com/inigomanuel/Computacion_Paralela_y_Distribuida/tree/main/04_Cuarta_Tarea