



Arquitectura de Computadores

- Pablo Rayón Zapater

- Fernando Pérez Ballesteros

- José María Fernández Práctica 3

Índice

I. Entregables ……….........................................................................3

A. Ejercicio I ...............................................….............................3

B. Ejercicio II ...............................….............................................4

C. Ejercicio III ..............................................…..............................5

C. Ejercicio IV ..............................................…..............................6

# Ejercicio 1.1

# **Explicar en qué consiste la comunicación colectiva, las primitivas que hay en MPI y las ventajas y desventajas de este tipo de comunicación.**

Es un tipo de comunicación que se diferencia en la punto a punto en que toman parte todos los procesos creados, se toma como referencia siempre un nodo maestro o raíz que es el que se encarga de enviar o recibir, en función del tipo de función de comunicación que sestemos tratando.

# Ejercicio 1.2

# **Implementar un programa donde el nodo 0 inicializa una variable con la frase “Hola mundo” lo envía al resto de nodos del comunicador.**

# 

Texto

Descripción generada automáticamente

Se ha implementado un programa en el que el nodo 0 pide un mensaje a el usuario, el cuál es el único que lo conoce y después lo envía a el resto de procesos y a si mismo incluido.

# Ejercicio 2

# **Implementar un programa donde el nodo 0 inicializa un array unidimensional asignando a cada valor su índice. Este array es dividido en partes, donde cada una de ellas será mandada a un proceso/nodo diferente. Después de que cada nodo haya recibido su porción de datos, los actualiza sumando a cada valor su rank. Por último, cada proceso envía su porción modificada al proceso root. Captura de pantalla de computadora Descripción generada automáticamente**

# **Imagen que contiene Texto Descripción generada automáticamente**

# En este ejercicio el proceso 0 inicializa un array asignándole a cada posición el valor de su número de índice para más tarde repartir en sub-arrays de tres elementos a cada uno de los programas, esto se ha decidido así porque el array que inicializa el proceso 0, rellena un array de el numero de procesos multiplicado por tres elementos. Más tarde estos mismos procesos reenvían los tres primeros elementos de su array, una vez han operado con él, al proceso 0, el cual acabara imprimiendo el resultado final.

# Ejercicio 3

# **Implementar un programa donde cada proceso inicializa un array de una dimensión, asignando a todos los elementos el valor de su rank+1. Después el proceso 0 (root) ejecuta dos operaciones de reducción (suma y después producto) sobre los arrays de todos los procesos.**

# 

# 

Es un caso similar a el anterior, la diferencia es que en lugar de que el proceso 0 inicialice el array, cada uno de los procesos inicializa su propio array. Más adelante, con la operación reduce, el proceso 0 recibe cada uno de estos array, para aplicarle una operación para cada uno de los elementos del array. El primer reduce aplica un sumatorio y el segundo un productorio.

Ejercicio 4

**Crear un programa que, haciendo uso de las funciones de comunicación colectiva que se considere necesario, calcule el factorial del número total de procesos. Esto es, si se ejecuta la aplicación con 4 procesos, uno de los procesos tiene que mostrar el resultado 24, si se ejecuta con 5, el resultado sería 120.**

****



En este ejercicio, se ha observado que para un caso particular del ejercicio anterior se cumple con lo requerido en el enunciado, esto es debido a que cada uno de los procesos rellena su array con un numero ascendente para cada proceso respectivamente, empezando desde 1. Como cada uno de estos sub-arrays contiene como valores el correspondiente a su rango de proceso, una vez apliquemos un productorio a cada uno de los arrays con la función reduce, conseguiremos hallar el factorial del número de procesos, lo realizará el proceso 0, al que todos los procesos le enviaran sus sub-arrays.