

**LEAD UNIVERSITY**

**COMPUTACION PARALELA Y DISTRIBUIDA**

# **TAREA 3: MEDICION DE LATENCIA Y COMUNICACIÓN COLECTIVA**

**II CUATRIMESTRE 2024**

**PROFESOR: JOHANSELL VILLALOBOS CUBILLO**

**ESTUDIANTE: WALTER BONILLA CAMACHO**

## 1. Descripción

### Programa A: estadísticas\_mpi.py

Objetivo: Calcular estadísticas globales (mínimo, máximo y promedio) de un arreglo de números distribuido entre múltiples procesos usando MPI.

#### Funcionamiento:

El proceso raíz (rank=0) genera un arreglo de 1,000,000 números aleatorios (entre 0 y 100).

Divide el arreglo en 4 partes iguales (250,000 elementos por proceso).

Cada proceso calcula sus estadísticas locales.

Usando operaciones MPI\_Reduce, se obtienen los resultados globales.

Finalmente, se reconstruye el arreglo original con MPI\_Gather y se verifica su precisión.

### Programa B: latencia\_p2p\_mpi.py

Objetivo: Medir la latencia (tiempo de comunicación) entre dos procesos usando operaciones punto a punto (MPI\_Send y MPI\_Recv).

#### Funcionamiento:

Proceso 0 envía un mensaje de 1 byte al Proceso 1, que lo devuelve inmediatamente.

Este ciclo se repite 10,000 veces para obtener un promedio preciso.

Se calcula la latencia de ida y vuelta y se estima la unidireccional.

## 2. Análisis de Resultados

### Resultados del Programa 1 (Estadísticas MPI)

Métrica	Valor MPI	Valor NumPy (Verificación)
Mínimo	0.0000	0.0000
Máximo	99.9998	99.9998
Promedio	49.9243	49.9243

#### Interpretación:

Los resultados de MPI coinciden exactamente con los de NumPy, lo que confirma que la distribución y reducción de datos funcionan correctamente.

El arreglo se reconstruyó sin errores (Gather exitoso), validando la integridad de los datos.

El uso de 4 procesos permitió paralelizar el cálculo eficientemente.

### Resultados del Programa 2 (Latencia Punto a Punto)

Métrica	Valor
---------	-------

Latencia (ida y vuelta)	5.45 $\mu$ s
-------------------------	--------------

Latencia estimada (unidireccional)	2.73 $\mu$ s
------------------------------------	--------------

Interpretación:

La latencia de 5.45  $\mu$ s (ida y vuelta) es típica para comunicaciones MPI en una misma máquina.

La latencia unidireccional estimada (2.73  $\mu$ s) es consistente con el rendimiento esperado en sistemas modernos.

Este tiempo incluye:

Overhead de MPI (gestión de mensajes).

Tiempo de envío/recibo entre procesos.

### 3. Conclusiones

#### Paralelización efectiva:

El primer programa demostró que MPI puede distribuir cálculos estadísticos sin pérdida de precisión.

La reconstrucción del arreglo (Gather) garantiza que los datos no se corrompen durante la comunicación.

#### Baja latencia en comunicaciones:

El segundo programa confirmó que MPI es eficiente para operaciones punto a punto, con latencias en el rango de microsegundos.

Esto es crucial para aplicaciones que requieren sincronización frecuente entre procesos (ej: simulaciones en tiempo real).

#### Recomendaciones:

Para problemas más grandes, aumentar el número de procesos (-np 8 o más) podría mejorar el rendimiento.

En redes distribuidas (varias computadoras), la latencia sería mayor debido al hardware de red.