



Práctica 3 de 3 – Introducción a MPI (Práctica Final)

Recordatorio previo

Esta práctica es la tercera y última parte del bloque de Introducción a MPI.

En ella aplicarás de forma integrada los principales conceptos ya practicados:

- Comunicación punto a punto (`MPI_Send``, `MPI_Recv``)
- Comunicación colectiva (`MPI_Scatter``, `MPI_Gather``, `MPI_Reduce``, `MPI_Bcast``)
- Sincronización (`MPI_Barrier``) y medición de tiempos (`MPI_Wtime``)
- Gestión de errores básica

Recuerda que un programa MPI típico tiene esta estructura:

1. Inicialización (`MPI_Init``)
2. Obtención del `rank`` y `size``
3. Comunicación o cálculo según el rol de cada proceso
4. Finalización (`MPI_Finalize``)

Planteamiento: Simulación distribuida de sensores

Debes implementar un sistema paralelo en el que se simula una red distribuida de sensores que recogen valores (por ejemplo, temperaturas o presiones).

El objetivo es calcular medias locales y detectar posibles valores críticos que activen una alerta global.

El objetivo es calcular medias locales y detectar posibles valores críticos que activen una alerta global. Actividades (guía sugerida)

1. **El proceso 0** generará aleatoriamente un array de valores enteros (ej. 5 por proceso).
2. Distribuye los valores entre procesos usando `MPI_Scatter``.
3. Cada proceso calcula la media local de sus valores.



4. Usa ``MPI_Gather`` para enviar las medias locales al proceso 0.
5. El proceso 0:
 1. Muestra todas las medias locales.
 2. Detecta si alguna supera un umbral crítico (ej. > 50).
 3. En caso afirmativo, lanza un mensaje de alerta e identifica los procesos responsables.
6. Opcional:
 4. Añade ``MPI_Barrier`` antes del análisis para sincronizar.
 5. Usa ``MPI_Wtime`` para medir el tiempo de ejecución.
 6. Añade control de errores: por ejemplo, que el número de elementos sea múltiplo del número de procesos.

Preguntas de reflexión

Responde de forma justificada las siguientes preguntas:

1. ¿Qué tipo de comunicación has usado en cada parte del programa?
2. ¿Hay algún tipo de desequilibrio en el trabajo de los procesos?
3. ¿Dónde podrías aplicar paralelismo adicional?
4. ¿Qué otras estrategias se te ocurren para lanzar alertas o combinar resultados?

Instrucciones de entrega final

Esta práctica se entrega de forma conjunta con las dos anteriores. Debes subir a Moodle un archivo ``zip`` que incluya:

- Código fuente (``c``) de esta práctica.
- Códigos fuente de las prácticas anteriores.
- Capturas de pantalla que demuestren la ejecución.
- Un documento ``rtf`` o ``pdf`` con:
 - Las respuestas a las preguntas de esta práctica.
 - Las reflexiones de las prácticas anteriores.
 - Comentarios personales si los deseas (opcional).



Rúbrica de evaluación conjunta (3 prácticas)

Criterio	Descripción	Puntos
Entrega completa	Incluye todas las prácticas, códigos y documentos solicitados.	3
Aplicación correcta de MPI	Usa adecuadamente las funciones de MPI vistas en clase.	2
Calidad del código	Códigos claros, funcionales y bien comentados.	2
Reflexión y análisis	Responde con criterio a las preguntas planteadas.	2
Organización	Buena presentación y estructura del archivo entregado.	1

Justificación Académica

Esta práctica final sintetiza los contenidos clave de la unidad de MPI trabajados en clase:

- Comunicación punto a punto y colectiva
- Distribución y recogida de datos
- Reducción y análisis global
- Sincronización, temporización y detección de errores

La actividad promueve la aplicación integrada de técnicas de paralelismo en memoria distribuida y refuerza competencias relacionadas con la computación de alto rendimiento, incluyendo diseño de soluciones escalables y análisis del comportamiento del sistema paralelo bajo diferentes configuraciones.