



UA

**Universidad
de Alicante**

ESCUELA POLITÉCNICA SUPERIOR

Introducción a MPI

Curso académico 2024 / 2025

Semestre 2

Grado en Ingeniería en Inteligencia Artificial

Grupo 1

Profesora: Ricardo Moreno Rodríguez

Alumno: Santiago Álvarez Geanta

ÍNDICE

Introducción	2
Ejercicio 1: Medición de Speed-up y Eficiencia	2
Código en OpenMP	2
Resultados obtenidos	2
Ejercicio 2 Comparación de Sincronización	3
Código en OpenMP	3
Resultados obtenidos (critical version)	3
Resultados obtenidos (atomic version)	3
Resultados obtenidos (reduction version)	4
Análisis y Conclusiones	4
Ejercicio 3 Balanceo de carga y Schedule	4
Código en OpenMP	4
Análisis y Conclusiones	5
Ejercicio 4: Análisis de Rendimiento y Ley de Amdahl	5
Ejercicio 5 (Opcional): Escalabilidad Débil	6
Código en OpenMP	6
Análisis y Conclusiones	7

Parte 1: Investigación Guiada sobre MPI

Cada estudiante deberá responder de manera individual a las siguientes cuestiones, integrando la información en un informe o presentación que formará parte de la entrega final.

1. ¿Qué es MPI?

MPI es un estándar de comunicación diseñado para facilitar la programación paralela y distribuida en sistemas de alto rendimiento. Su principal objetivo es permitir la comunicación eficiente entre procesos que se ejecutan en diferentes nodos de un clúster o supercomputadora. Es ampliamente utilizado en simulaciones científicas, análisis de datos y modelado matemático debido a su capacidad para optimizar el procesamiento en paralelo.

2. ¿MPI es un lenguaje de programación o una biblioteca?

MPI no es un lenguaje de programación, sino una biblioteca estándar que proporciona funciones para la comunicación entre procesos. Es compatible con varios lenguajes de programación, entre los que destacan:

- C y C++, los lenguajes más utilizados con MPI debido a su eficiencia y compatibilidad con los compiladores de alto rendimiento.
- Fortran, especialmente popular en aplicaciones científicas.
- Python, mediante bibliotecas como *mpi4py*, que facilita la integración de MPI con este lenguaje.
- Java, aunque su uso con MPI es menos común, existen implementaciones que permiten su integración.

3. ¿Quién desarrolla y mantiene MPI?

MPI es un estándar definido y mantenido por el *MPI Forum*, un consorcio de expertos en computación paralela que incluye universidades, centros de supercomputación y empresas tecnológicas. Algunas de las organizaciones que participan en su desarrollo son:

- Los Alamos National Laboratory
- Argonne National Laboratory
- IBM y Intel, entre otras empresas tecnológicas.

4. Versiones de MPI

MPI ha evolucionado con el tiempo, incorporando mejoras en cada versión:

- MPI-1: Introdujo el modelo básico de comunicación entre procesos.
- MPI-2: Añadió soporte para operaciones de entrada/salida en paralelo y memoria compartida.
- MPI-3: Mejoras en la gestión de procesos y nuevos tipos de comunicación.
- MPI-4: La versión más reciente, con optimizaciones para entornos modernos de supercomputación.

5. Principales usos de MPI

MPI es ampliamente utilizado en distintas áreas, incluyendo:

- Supercomputación: Facilita la ejecución de simulaciones complejas en clústeres y supercomputadoras.
- Inteligencia artificial y Machine Learning: Mejora el rendimiento de los algoritmos distribuidos.
- Análisis de datos a gran escala: Es clave en instituciones como el CERN para el procesamiento de datos del *Gran Colisionador de Hadrones*.

Algunas de las organizaciones que utilizan MPI incluyen:

- NASA: Para simulaciones aeroespaciales.
 - CERN: En el procesamiento de datos científicos.
 - Barcelona Supercomputing Center: Aplicaciones en climatología y modelado.
-

Parte 2: Elaboración del Informe sobre Preparación de Entornos

Message Passing Interface (MPI) es un estándar de comunicación para la programación paralela y distribuida. Su implementación en distintos sistemas operativos permite ejecutar programas en múltiples procesadores, facilitando el desarrollo de aplicaciones de alto rendimiento. Este informe detalla los pasos necesarios para instalar y configurar MPI en Ubuntu, proporcionando un entorno funcional para la programación en C y C++.

1. Resumen de opciones de instalación

A continuación, se presentan las opciones más utilizadas para instalar MPI en distintos sistemas operativos:

Sistema Operativo	Implementaciones de MPI	Método de Instalación
Windows	MS-MPI, OpenMPI en WSL	Instalador oficial, subsistema Linux
Linux	OpenMPI, MPICH	Gestores de paquetes (apt, yum, pacman)
MacOS	OpenMPI	Homebrew

2. Desarrollo detallado de un caso específico



2.1. Descarga e instalación del entorno necesario

Ubuntu ofrece soporte para OpenMPI y MPICH. En este informe se utiliza OpenMPI, una de las implementaciones más populares y compatibles con diversos entornos.

Pasos para la instalación:

1. Actualizar el sistema:

```
sant_vz6@santvz6-IdeaPad-Gaming-3-15ACH6:~$ sudo apt update && sudo apt upgrade -y
```

2. Instalar OpenMPI:

```
sant_vz6@santvz6-IdeaPad-Gaming-3-15ACH6:~$ sudo apt install -y openmpi-bin openmpi-common openmpi-doc libopenmpi-dev
```

3. Verificar la instalación:

```
sant_vz6@santvz6-IdeaPad-Gaming-3-15ACH6:~$ mpirun --version
mpirun (Open MPI) 4.1.6
```

2.2. Configuración de compiladores y bibliotecas

Para garantizar la correcta ejecución de programas MPI, es necesario configurar los compiladores y bibliotecas.

Pasos de configuración:

1. Comprobar la instalación del compilador:

```
sant_vz6@santvz6-IdeaPad-Gaming-3-15ACH6:~$ mpicc --version
gcc (Ubuntu 13.3.0-6ubuntu2~24.04) 13.3.0
Copyright (C) 2023 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

2. Configurar variables de entorno:

- Agregar las siguientes líneas al archivo ~/.bashrc o ~/.profile:

```
sant_vz6@santvz6-IdeaPad-Gaming-3-15ACH6:~$ export PATH=/usr/lib/openmpi/bin:$PATH
sant_vz6@santvz6-IdeaPad-Gaming-3-15ACH6:~$ export LD_LIBRARY_PATH=/usr/lib/openmpi/lib:$LD_LIBRARY_PATH
```

Esto permite que los ejecutables de OpenMPI (como **mpirun** y **mpicc**) se puedan ejecutar sin necesidad de especificar su ruta completa. Además, la segunda línea, le indica al sistema dónde buscar las bibliotecas compartidas de OpenMPI cuando se ejecutan programas que dependen de ellas.

- Aplicar los cambios:

```
sant_vz6@santvz6-IdeaPad-Gaming-3-15ACH6:~$ source ~/.bashrc
```

3. Verificar la configuración:

```
sant_vz6@santvz6-IdeaPad-Gaming-3-15ACH6:~$ which mpirun
/usr/bin/mpirun
```

2.3. Ejemplo de ejecución de un programa MPI básico

Para verificar que la instalación ha sido exitosa, se debe compilar y ejecutar un programa simple en MPI. Este programa debe demostrar la correcta comunicación entre procesos y confirmar que el entorno está listo para su uso.

```
1  #include <mpi.h>
2  #include <stdio.h>
3
4  int main(int argc, char** argv) {
5
6      MPI_Init(&argc, &argv);
7      int rank, size;
8
9      MPI_Comm_rank(MPI_COMM_WORLD, &rank);
10     MPI_Comm_size(MPI_COMM_WORLD, &size);
11
12     printf("Hola desde el proceso %d de %d!\n", rank, size);
13     MPI_Finalize();
14
15     return 0;
16 }
```

Compilación y ejecución:

1. Compilar el programa:

```
sant_vz6@santvz6-IdeaPad-Gaming-3-15ACH6:~/Escritorio/UA-2/2 CUATRI/CAR/P9$ mpicc basic_mpi.c -o basic
sant_vz6@santvz6-IdeaPad-Gaming-3-15ACH6:~/Escritorio/UA-2/2 CUATRI/CAR/P9$ ls
basic  basic_mpi.c
```

2. Ejecutar el programa con 4 procesos:

```
sant_vz6@santvz6-IdeaPad-Gaming-3-15ACH6: ~/Escritorio/UA-2/2 CUATRI/CAR/P9$ mpirun -np 4 ./basic
```

3. Salida esperada:

```
Hola desde el proceso 2 de 4!  
Hola desde el proceso 0 de 4!  
Hola desde el proceso 1 de 4!  
Hola desde el proceso 3 de 4!
```

3. Conclusiones

La instalación y configuración de OpenMPI en Ubuntu es un proceso relativamente sencillo utilizando los repositorios oficiales de APT. La correcta configuración de las variables de entorno y la verificación del compilador aseguran que el entorno esté listo para desarrollar aplicaciones paralelas con MPI. La ejecución de un programa de prueba confirma la funcionalidad del entorno.

Este informe proporciona una guía detallada para configurar MPI en Ubuntu, facilitando el desarrollo de programas paralelos en C y C++.