PRÁCTICA DE LABORATORIO / TALLER Nro. 2

Carrera de Computación

A. DATOS INFORMATIVOS		
Asignatura: Algoritmos, Análisis y Programación Paralela	**Ciclo / Semestre:** Séptimo	**Paralelo:** A
Docente: Roberth Gustavo Figueroa Díaz	**Período Académico:** Septiembre 2025 – Febrero 2026	
Alumnos: ● Gerardo Herrera ● Alexis Ludeña ● Luis Medina ● Byron Jimenez ● Nayely Ramirez		

B. INFORMACIÓN GENERAL	
Unidad: 2. Algoritmos naturalmente paralelos y programación concurrente	
Tema: Algoritmos naturalmente paralelos usando OpenMP y MPI	
Fecha: Loja, 12 de Diciembre de 2025	**Nro. horas:** 3h
Objetivos:	
● Algoritmos naturalmente paralelos y programación concurrente	
Corresponde al resultado de aprendizaje:	
Identifica tareas independientes en un programa que podría ser paralelizado, bajo los principios de solidaridad, transparencia, responsabilidad y honestidad.	
Recursos y/o materiales:	
● Computador. – Computadora con acceso a internet. ● Navegador. – Revisión de información por parte del estudiante en la web. ● Bibliografía. – Textos físicos, virtuales y materiales facilitados en el Sílabo. ● Biblioteca virtual. – Acceso a recursos digitales como libros y artículos. ● EVA . – Entorno virtual de aprendizaje de la UNL.	

C. DESARROLLO
Instrucciones: 1. Revisar el contenido facilitado por el profesor: diapositivas, libro base, sílabo de la asignatura, recursos digitales, use el libro base e información complementaria; así como, internet para reforzar los contenidos estudiados. 2. Revise detalladamente el material indicado en clase. 3. Realizar la lectura del texto base y complementarios sobre las temáticas abordadas. 4. Para el desarrollo de los ejercicios puede usarse las herramientas de su interés tales como lenguaje Python, Java o C++, a su elección.
Preguntas a responder:

1. **Elabore un un informe de la actividad realizada en grupo. Se debe utilizar los archivos adjuntos y verificar el funcionamiento en los ambientes:**

   a. **Secuencial (local)**

Para la ejecución del programa de multiplicación de matrices utilizando la configuración local y secuencial, se documentan los resultados obtenidos (Tabla 1) que servirán como la línea base de rendimiento ya que establece la medición base de tiempo para las comparaciones con las implementaciones de memoria compartida (OpenMP) y memoria distribuida (MPI).

Nro. Procesos	Nro. Núcleo	Tiempo	Checksum
16	8	7.50043 segundos	-336.504

Tabla 1. Resultados ejecución secuencial

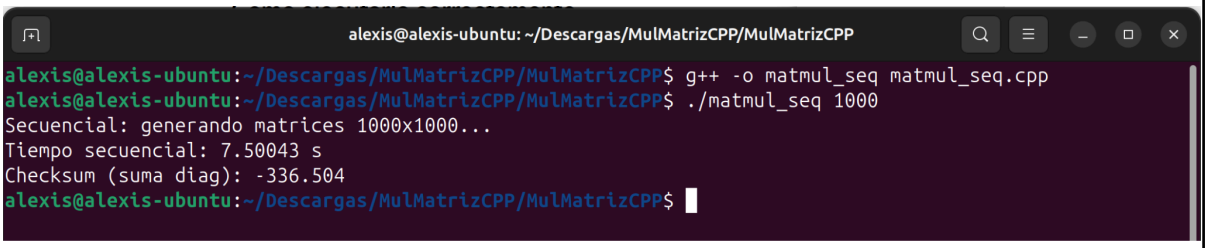


Figura 1. Resultados de la ejecución secuencial.

   b. **Memoria compartida (local)**

Para la ejecución del problema de multiplicación de matrices utilizando memoria compartida, se hizo uso de la librería OpenMP (Open multi-process), el uso de dicha librería facilitó el paralelismo de la aplicación sin manejar hilos manualmente ideal para CPU multinúcleo, los hilos comparten variables y memoria.

En el ejercicio propuesto, OpenMP permite dividir el trabajo de la multiplicación de matrices entre varios hilos, ejecutando la multiplicación en paralelo.

El objetivo principal de la ejecución con memoria compartida es acelerar el cálculo aprovechando múltiples núcleos del procesador, haciendo que varios hilos trabajen simultáneamente sobre un mismo espacio de memoria.

Nro. Procesos (Hilos)	Nro. Núcleo	Tiempo	Checksum
16	8	2.30024 segundos	-336.504

Tabla 2. Resultados ejecución Memoria Compartida

Tal como se puede observar en la tabla 2, los resultados obtenidos evidencia que la ejecución del programa utilizando memoria compartida presenta un incremento significativo en la velocidad de procesamiento en comparación con los resultados de la ejecución del programa secuencial, manteniendo el mismo tamaño de las matrices se registra una reducción del aproximada del 60% en el tipo de ejecución, lo cual se debe al aprovechamiento del paralelismo mediante múltiples hilos, permitiendo que las operaciones de multiplicación se ejecuten de forma simultánea y sea más eficiente sobre los recursos del sistema.

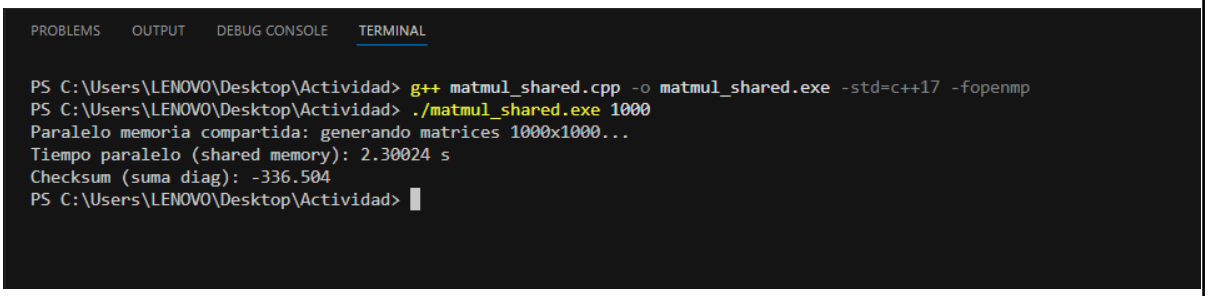


Figura 2. Resultado de la ejecución

c. Memoria distribuida (clúster)

La implementación del clúster se llevó a cabo sobre el sistema operativo Linux, específicamente utilizando la distribución Ubuntu, seleccionada por su estabilidad y compatibilidad nativa con entornos de computación distribuida.

Para garantizar el desarrollo, la administración y el correcto funcionamiento del clúster, se integraron las siguientes tecnologías y herramientas:

- SSH (Secure Shell): Empleado para establecer acceso remoto seguro y gestionar la administración de los nodos del clúster.
- MPI (Message Passing Interface): Estándar utilizado para la programación paralela, facilitando la comunicación y coordinación eficiente entre los procesos distribuidos.
- GCC (GNU Compiler Collection): Compilador usado para la construcción de los programas en C/C++ dentro del entorno MPI.
- Herramientas de red y monitoreo: Se utilizó ping para verificar la conectividad entre nodos , scp para la transferencia segura de archivos y htop/top para el monitoreo en tiempo real del uso de CPU, memoria y gestión de procesos.

A continuación, se presentan los tiempos de ejecución obtenidos al distribuir la carga de trabajo entre los nodos del clúster (Maestro y Esclavo).

Pc	Tipo Nodo	Proceso	Tiempo (segundos)
Luis	Maestro	1	0:14.39
Luis	Maestro	2	0:14.41
Luis	Maestro	3	0:14.42
Luis	Maestro	4	0:14.42
Luis	Maestro	5	0:14.41
Luis	Maestro	6	0:14.41
Luis	Maestro	7	0:14.41
Luis	Maestro	8	0:14.42
Bayron	Esclavo	9	0:40.71
Bayron	Esclavo	10	0:39.85
Bayron	Esclavo	11	0:38.51
Bayron	Esclavo	12	0:38.21

Tabla 3. Resultados ejecución memoria distribuida

Vista nodo Maestro:

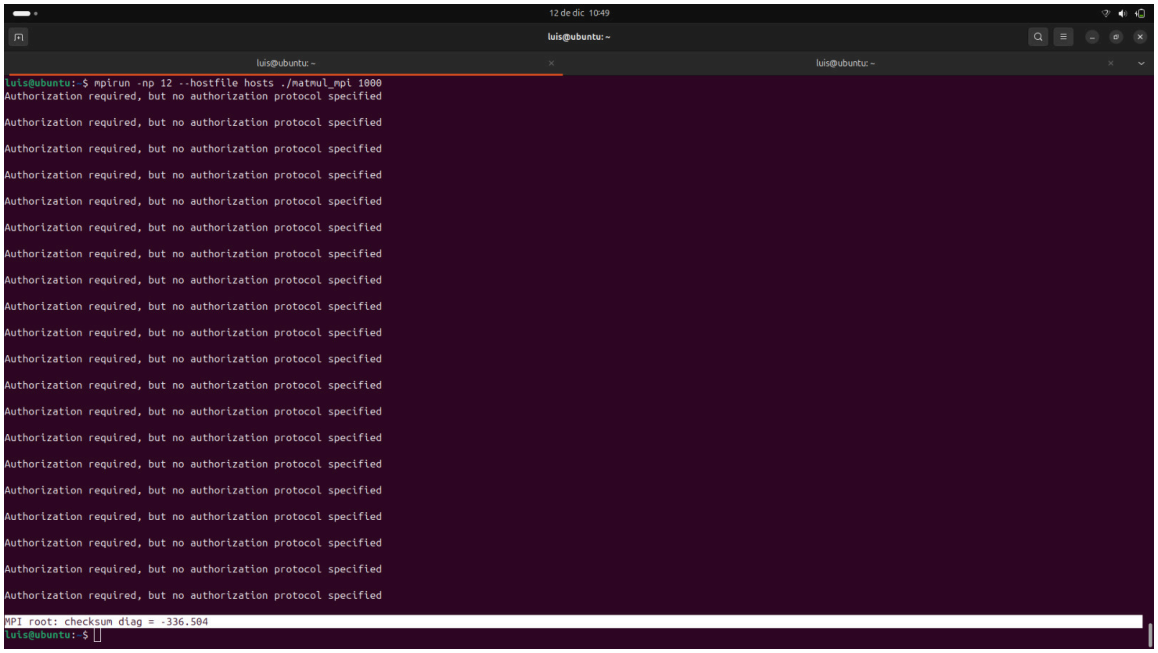


Figura 3. Resultados ejecución de la memoria distribuida

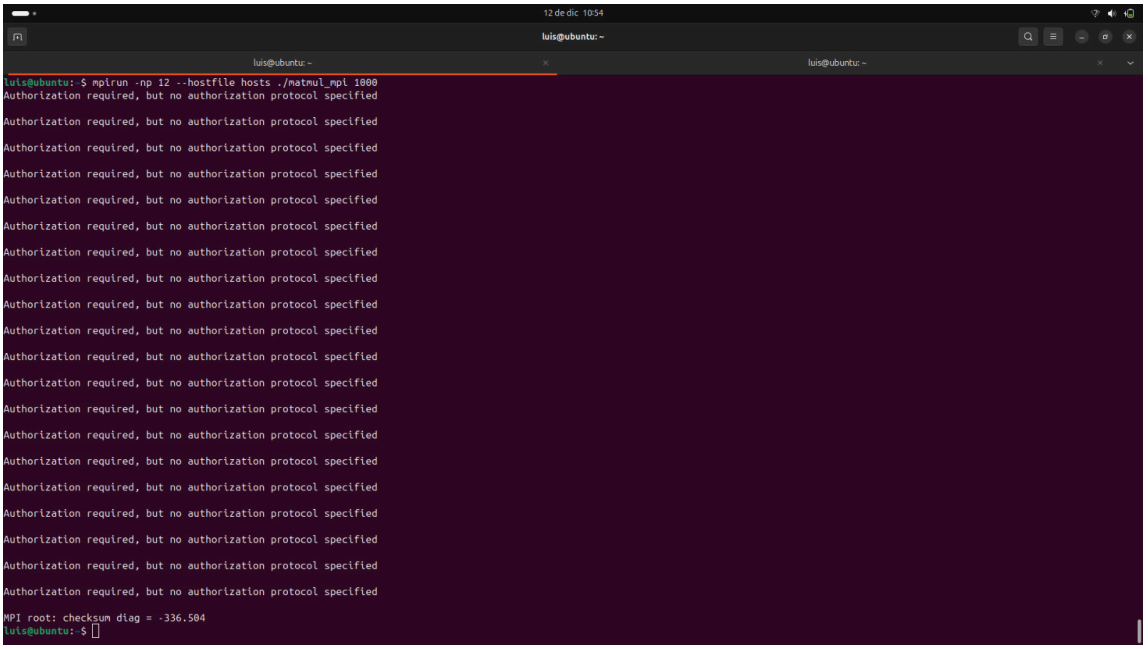


Figura 4. Procesos del nodo maestro de matmul

Como se evidencia en las Figuras 3 y 4, el nodo maestro gestionó la orquestación del proceso mediante el comando mpirun. Se observa que el *checksum* de la diagonal (-336.504) es consistente con las ejecuciones secuenciales y de memoria compartida, validando la corrección del cálculo distribuido.

Vista del nodo esclavo:

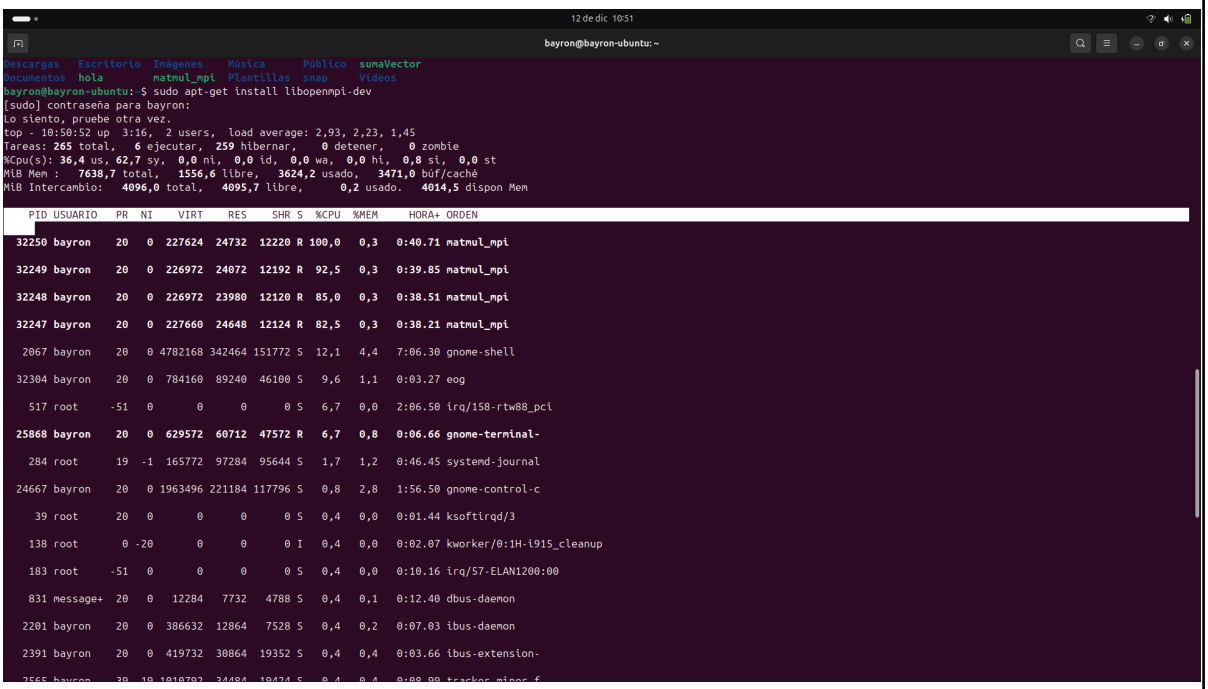


Figura 5. Procesos del nodo esclavo de matmul

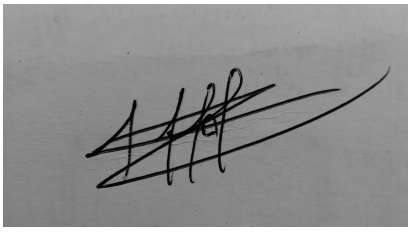
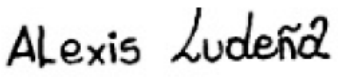
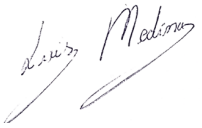
En la figura 5 se puede observar el funcionamiento del clúster desde la parte del nodo esclavo. En el primer comando se enlistan los documentos en el directorio y se observa a "*sumaVector*" y "*matmul_mpi*" que fueron enviados por el nodo maestro a través de *ssh*.
También se realizó la instalación de *libopenmpi-dev* para habilitar la ejecución de archivos que utilicen MPI. Finalmente con el comando top se visualiza los procesos en ejecución y entre los primeros destaca "*matmul_mpi*" que fue ejecutado por el maestro en este nodo.

Conclusiones:

- La práctica permitió validar la utilidad de la programación paralela tanto en memoria compartida (OpenMP) como distribuida (MPI), utilizando los códigos desarrollados para medir y comparar el rendimiento en cada escenario.
- Mediante la implementación de los códigos matmul_seq.cpp, matmul_openmp.cpp y matmul_mpi.cpp, se comprobó que la paralelización con OpenMP reduce considerablemente el tiempo de ejecución frente al enfoque secuencial (7.50043 segundos - Secuencial a 2.30024 segundos - OpenMP), aprovechando múltiples núcleos de procesador.
- El uso del código MPI permitió distribuir la carga de trabajo de la multiplicación de matrices entre los nodos Maestro y Esclavo, demostrando la eficacia de la librería para coordinar y optimizar el procesamiento en sistemas de múltiples nodos, confirmando la utilidad del MPI en arquitecturas de computación distribuida.

D. RÚBRICA DE EVALUACIÓN	
Informe de trabajo: ● Contenido: pertinente y concreto. ● Estructura y organización: Elementos vinculados y estructurados coherentemente. ● Originalidad y creatividad: trabajo inédito, presentación de nuevas ideas.	2 puntos
Resolución de Preguntas: ● Proceso de resolución de preguntas: con originalidad y creatividad	6 puntos
Conclusiones: ● Redacción ● Originalidad y creatividad: conclusiones inéditas en base a su experiencia y objetivos	2 puntos
Total	10 puntos

(Ponderado en Aprendizaje Práctico Experimental)	

E. FIRMAS DE RESPONSABILIDAD:	
Estudiante(s):	**Firma**
Gerardo Israel Herrera Campoverde	
Byron Alejandro Jimenez Cango	
Alexis Gradyn Ludeña Cueva	
Luis Alberto Medina Chamba	
Nayely Cruzcaya Ramirez Herrera