



UNIVERSIDAD NACIONAL DE LOJA

CARRERA DE COMPUTACIÓN

Algoritmos, Análisis y Programación Paralela

Septiembre 2025 – Febrero 2026

Actividad de Aprendizaje Autónomo # 2

Tema: Revisión rápida de procesamiento secuencial y paralelo.

Integrantes:

- Luis Alberto Medina Chamba.
- Gerardo Israel Herrera Campoverde.
- Bayron Alejandro Jimenez Cango.
- Alexis Grady Ludeña Cueva.
- Nayely Cruzcaya Ramirez Herrera.

Fecha: 29 de diciembre de 2025.

Objetivo: Revisar, analizar y comprender el siguiente ejercicio sobre y paralelo.

A. Actividades

Se debe realizar lo siguiente:

1. Configure un clúster con las tecnologías a elección del grupo para evaluar algoritmos paralelos.
2. Elabore un diagrama con el diseño de la arquitectura del clúster elaborado.

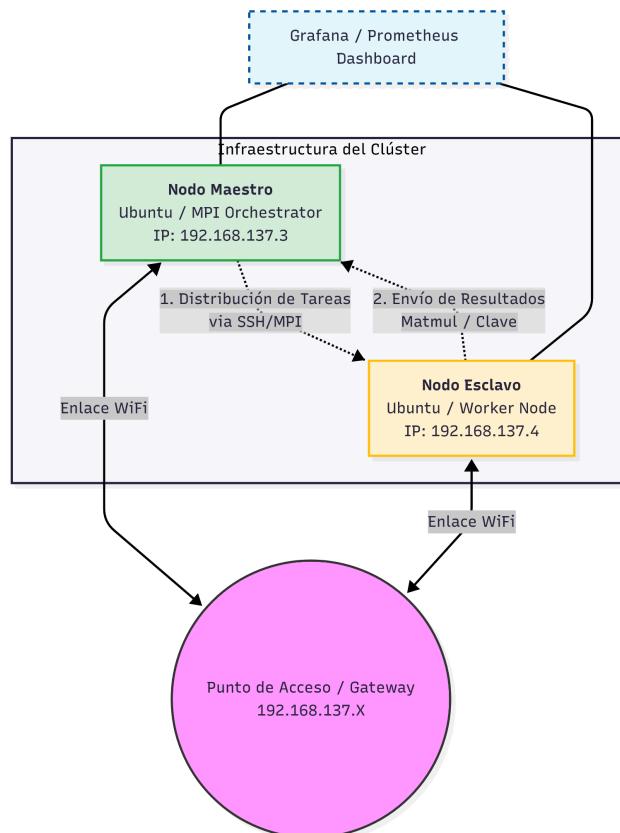


Fig. 1 - Estructura arquitectónica del cluster desarrollado.

La arquitectura presentada (Fig. 1) corresponde a un clúster de computación distribuida con modelo maestro–esclavo, interconectado mediante una red inalámbrica a través de un punto de acceso.

Punto de Acceso

El punto de acceso (red 192.168.137.X) actúa como el elemento central de conectividad, permitiendo que los nodos del clúster se comuniquen dentro de la misma red local.

Ambos nodos se conectan a este punto mediante enlace WiFi, quedando en la misma subred, lo que garantiza comunicación directa y estable.

Nodo Maestro (192.168.137.3)

El nodo maestro, ejecutando Ubuntu, cumple el rol de orquestador MPI. Sus funciones principales son:

- Inicializar la ejecución de los programas paralelos.
- Distribuir las tareas a los nodos esclavos mediante SSH y MPI.
- Coordinar la ejecución de los procesos paralelos.
- Recopilar y consolidar los resultados enviados por los nodos esclavos.
- Este nodo es el encargado de controlar el flujo general del cómputo distribuido.

Nodo Esclavo (192.168.137.4)

El nodo esclavo, también bajo Ubuntu, funciona como nodo de trabajo (worker). Sus responsabilidades incluyen:

- Recibir las tareas asignadas por el nodo maestro.
- Ejecutar el cálculo correspondiente.
- Enviar los resultados parciales de vuelta al nodo maestro.
- Este nodo no toma decisiones de control, sino que se enfoca en el procesamiento.

Comunicación

El flujo de comunicación se desarrolla en dos fases principales:

- Distribución de tareas:
El nodo maestro envía las tareas al nodo esclavo usando SSH/MPI, aprovechando la conectividad establecida por el punto de acceso.
- Envío de resultados:
El nodo esclavo devuelve los resultados parciales al nodo maestro, quien se encarga de integrarlos para obtener el resultado final.

Monitoreo y control

En la parte superior de la arquitectura se muestra un dashboard de Grafana utilizado para:

- Monitorear el uso de CPU, memoria y red.
- Observar el comportamiento del clúster durante la ejecución paralela.
- Analizar el rendimiento de los nodos en tiempo real.

Este componente no participa directamente en el cómputo, pero es clave para la supervisión y análisis del desempeño.

3. Ejecute un ejemplo a criterio del grupo para revisar su funcionamiento.

Ejecución del algoritmo de Fuerza Bruta

El algoritmo de Fuerza Bruta Alfanumérico es un método de ataque o resolución de problemas (como la búsqueda de una contraseña o clave de cifrado) que funciona probando sistemáticamente cada posible combinación de caracteres (letras y números) hasta encontrar la correcta. Este algoritmo se implementó para distribuir la carga de trabajo (el espacio total de claves a probar) entre los diferentes nodos del clúster, permitiendo que cada nodo explore una porción del espacio de búsqueda de forma simultánea. Esto reduce significativamente el tiempo total necesario para encontrar la solución.

Este ejemplo fue ejecutado adicionalmente al algoritmo presentado en la sección de resultados (multiplicación de matrices) para la verificación del correcto funcionamiento del clúster y para que sirva como guía para el trabajo de fin de asignatura.

Fig. 2 - Ejecución del algoritmo de Fuerza Bruta Alfanumérico.

```
luis@ubuntu:~$ top - 10:44:11 up 1:25, 1 user, load average: 1,31, 2,69, 3,56
Tareas: 493 total, 9 ejecutar, 484 hibernar, 0 detener, 0 zombie
NCpu(s): 3,7 us, 0,5 sy, 0,0 ni, 95,8 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
MLB Mem : 15675,1 total, 9181,6 libre, 3554,4 usado, 3883,5 búf/caché
MLB Intercambio: 4096,0 total, 4095,7 libre, 0,2 usado. 12120,7 dispon Mem

PID USUARIO PR NI VIRT RES SHR S %CPU %MEM HORA+ ORDEN
20411 luis 20 0 27616 1916 1776 R 12,6 0,0 0:00:38 fuerzaBrutaAlfa
20415 luis 20 0 27616 1984 1844 R 10,9 0,0 0:00:33 fuerzaBrutaAlfa
20403 luis 20 0 27616 1984 1844 R 10,6 0,0 0:00:32 fuerzaBrutaAlfa
20407 luis 20 0 27616 1916 1776 R 10,6 0,0 0:00:32 fuerzaBrutaAlfa
20396 luis 20 0 4699176 25756 17792 S 6,0 0,2 0:00:18 mpirun
20419 luis 20 0 27616 1988 1844 R 6,0 0,0 0:00:18 fuerzaBrutaAlfa
20424 luis 20 0 27616 1984 1844 R 4,3 0,0 0:00:13 fuerzaBrutaAlfa
20402 luis 20 0 14700 8632 7276 S 3,3 0,1 0:00:10 ssh
20423 luis 20 0 27616 1916 1780 R 3,3 0,0 0:00:10 fuerzaBrutaAlfa
20431 luis 20 0 27616 1988 1844 R 3,0 0,0 0:00:09 fuerzaBrutaAlfa
2459 luis 20 0 24,9g 1808028 118692 S 2,6 1,1 5:42.22 Xorg
2780 luis 20 0 4793800 358700 152052 S 2,3 2,2 8:24.59 gnome-shell
18912 luis 20 0 842332 57496 44016 S 1,3 0,4 0:03:11 gnome-terminal-
347 root -51 0 0 0 0 D 1,0 0,0 0:33:18 irq/152-VEN_06CB:00
153 root -51 0 0 0 0 S 0,7 0,0 0:43:95 irq/9-acpi
1126 root 20 0 422536 10992 9824 S 0,7 0,1 0:23:07 thermald
7006 grafana 20 0 1880884 308692 187232 S 0,7 1,9 0:42:36 grafana
13248 luis 20 0 8432 6664 5196 S 0,7 0,0 0:00:22 ssh-agent
20294 luis 20 0 12236 6252 3924 R 0,7 0,0 0:00:17 top
75 root rt 0 0 0 0 S 0,3 0,0 0:01:27 migration/15
2324 luis -9 -11 132372 30824 11064 S 0,3 0,2 0:01:84 pipewire-pulse
2816 luis 20 0 1213800 133840 107396 S 0,3 0,8 0:19:18 mutter-x11-fram
2875 luis 20 0 386656 12948 7452 S 0,3 0,1 0:16:69 ibus-daemon
3024 luis 20 0 421800 34948 20528 S 0,3 0,2 0:04:31 ibus-extension-
3216 luis 20 0 234556 7896 7096 S 0,3 0,0 0:03:77 ibus-engine-sin
18722 root 20 0 0 0 0 I 0,3 0,0 0:00:16 kworker/u80:0-events_unbound
1 root 20 0 23832 14944 9400 S 0,0 0,1 0:08:58 systemd
2 root 20 0 0 0 0 S 0,0 0,0 0:00:03 kthreadadd
3 root 20 0 0 0 0 S 0,0 0,0 0:00:00 pool_workqueue_release
4 root 0 -20 0 0 0 I 0,0 0,0 0:00:00 kworker/R-rcu_gp
5 root 0 -20 0 0 0 I 0,0 0,0 0:00:00 kworker/R-sync_wq
6 root 0 -20 0 0 0 I 0,0 0,0 0:00:00 kworker/R-kvfree_rcu_reclaim
7 root 0 -20 0 0 0 I 0,0 0,0 0:00:00 kworker/R-slub_flushhw
8 root 0 -20 0 0 0 I 0,0 0,0 0:00:00 kworker/R-netns
10 root 0 -20 0 0 0 I 0,0 0,0 0:00:00 kworker/0:0h-events_highpri
```

Fig. 3 - Monitorización de recursos mediante el comando top.

```

19 de dic 10:43
bayron@bayron-ubuntu:~/node_exporter-1.7.0.linux-amd64
```

top - 10:43:16 up 2:59, 2 users, load average: 2,98, 2,09, 2,68
Tareas: 302 total, 5 ejecutar, 296 hibernar, 0 detener, 1 zombie
%Cpu(s): 99,2 us, 0,7 sy, 0,0 ni, 0,0 id, 0,0 wa, 0,0 hi, 0,1 si, 0,0 st
MiB Mem : 7638,7 total, 308,9 libre, 5515,3 usado, 3054,3 búf/caché
MiB Intercambio: 4096,0 total, 3877,8 libre, 218,2 usado, 2123,5 dispon Mem

PID USUARIO	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	HORA+ ORDEN
24056 bayron	20	0	27612	2064	1928 R	108,3	0,0	0,0	0:07.12 fuerzaBrutalAlfa
24051 bayron	20	0	27612	2068	1928 R	96,4	0,0	0,0	0:07.80 fuerzaBrutalAlfa
24055 bayron	20	0	27612	2064	1928 R	95,7	0,0	0,0	0:07.30 fuerzaBrutalAlfa
24060 bayron	20	0	27612	2004	1864 R	86,4	0,0	0,0	0:06.51 fuerzaBrutalAlfa
2647 bayron	20	0	4514168	237468	95708 S	2,0	3,0	19:53.89 gnome-shell	
283 root	19	-1	91812	43404	42488 S	1,7	0,6	12:13.71 systemd-journal	
12898 bayron	20	0	3840344	325348	164900 S	1,3	4,2	0:55.03 Isolated Web Co	
595 root	-51	0	0	0	0 S	0,7	0,0	2:47.05 irq/158-rtw88_pci	
931 root	20	0	334880	19036	15448 S	0,7	0,2	0:13.75 NetworkManager	
2214 bayron	20	0	541180	10696	9656 S	0,7	0,1	0:06.13 gsd-sharing	
4825 bayron	20	0	642672	54368	41184 S	0,7	0,7	0:31.27 gnome-terminal-	
9948 bayron	20	0	3830884	312624	181488 S	0,7	4,0	1:59.38 Isolated Web Co	
23133 bayron	20	0	12240	6276	3976 R	0,7	0,1	0:01.36 top	
826 message+	20	0	12464	7052	4612 S	0,3	0,1	0:21.64 dbus-daemon	
932 root	20	0	18264	11984	10348 S	0,3	0,2	0:05.66 wpa_supplicant	
2770 bayron	20	0	13,7g	686484	254756 S	0,3	8,8	57:59.21 firefox	
9237 bayron	20	0	3805720	291076	181100 S	0,3	3,7	1:27.78 Isolated Web Co	
9534 bayron	20	0	3492204	86736	70356 S	0,3	1,1	0:03.99 Isolated Web Co	
9574 bayron	20	0	3490161	82980	66964 S	0,3	1,1	0:04.81 Isolated Web Co	
11286 bayron	20	0	2685136	128564	64064 S	0,3	1,6	0:29.70 gnome-control-c	
13230 bayron	20	0	3972884	483716	106624 S	0,3	6,2	4:56.62 Isolated Web Co	
20650 root	20	0	0	0	0 I	0,3	0,0	0:00.56 kworker/0:1-events	
20281 root	20	0	0	0	0 I	0,3	0,0	0:01.50 kworker/u16:4-events_unbound	
21762 root	20	0	0	0	0 I	0,3	0,0	0:00.54 kworker/u16:0-events_unbound	
1 root	20	0	23528	14824	9644 S	0,0	0,2	0:28.48 systemd	
2 root	20	0	0	0	0 S	0,0	0,0	0:00.01 kthreadread	
3 root	20	0	0	0	0 S	0,0	0,0	0:00.00 pool_workqueue_release	
4 root	0	20	0	0	0 I	0,0	0,0	0:00.00 kworker/R-rcu_gp	
5 root	0	20	0	0	0 I	0,0	0,0	0:00.00 kworker/R-sync_wq	
6 root	0	20	0	0	0 I	0,0	0,0	0:00.00 kworker/R-kvfree_rcu_reclaim	
7 root	0	20	0	0	0 I	0,0	0,0	0:00.00 kworker/R-slub_flushwq	
8 root	0	20	0	0	0 I	0,0	0,0	0:00.00 kworker/R-netns	
10 root	0	20	0	0	0 I	0,0	0,0	0:00.00 kworker/0:H-events_highpri	
13 root	0	20	0	0	0 I	0,0	0,0	0:00.00 kworker/R-mm_percpu_wq	
14 root	20	0	0	0	0 I	0,0	0,0	0:00.00 rCU_tasks_kthread	
15 root	20	0	0	0	0 I	0,0	0,0	0:00.00 rCU_tasks_rude_kthread	
16 root	20	0	0	0	0 I	0,0	0,0	0:00.00 rCU_tasks_trace_kthread	
17 root	20	0	0	0	0 S	0,0	0,0	0:00.95 ksoftirqd/0	

Fig. 4 - Monitorización de recursos mediante el comando top (nodo esclavo).

4. Elabore un informe explicativo con las capturas e imágenes del funcionamiento con las tecnologías, versiones, librerías usadas, entre otras; y describa lo relevante en la configuración y funcionamiento correcto de su clúster.

Descripción General del Cluster

El cluster implementado corresponde a un cluster de cómputo (computadoras), utilizando memoria compartida, diseñado para la experimentación y análisis de algoritmos paralelos, el principal objetivo del cluster es distribuir la carga de trabajo entre múltiples nodos interconectados (computadoras), logrando ejecutar aplicaciones paralelas de manera eficiente y mucho más veloz frente a la ejecución secuencial.

La arquitectura del cluster está compuesta por dos tipos de nodos principales:

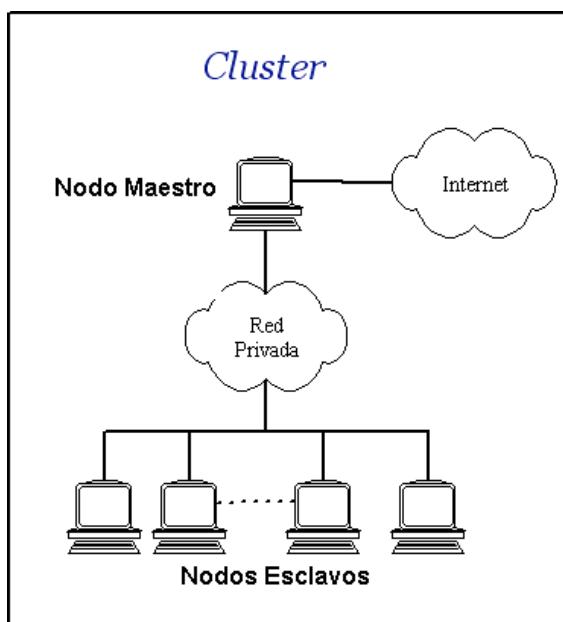


Fig. 5 - Arquitectura Cluster Maestro-Esclavo

- Nodo maestro: encargado de la coordinación general del cluster, la asignación y distribución de tareas, la inicialización de los procesos paralelos y la recuperación de los resultados de la solución.

- Nodo esclavo: Responsable de ejecutar las tareas asignadas por el nodo maestro, participando activamente en el procesamiento distribuido.

Ambos nodos se encuentran conectados en una misma red local, la cual fue configurada mediante un punto de acceso, asegurando la comunicación entre las máquinas del clúster.

El cluster fue utilizado para la ejecución de aplicaciones paralelas, el problema que presentamos para demostrar el funcionamiento del cluster fue la multiplicación de matrices.

Configuración General del Cluster

La configuración general del cluster se hizo en un entorno linux, se seleccionó este sistema operativo debido a la alta cantidad de herramientas y librerías especializadas en el entorno de comunicación y computación de alto rendimiento.

La arquitectura del cluster fue Maestro-Esclavo, para este caso la ejecución se hizo en dos nodos, en una red local, la comunicación de los nodos se hizo mediante SSH y MPI, y la distribución de la carga fue mediante MPI Scatter y Gather.

El nodo maestro actúa como el punto central de control, ejecutando el comando *mpirun* para iniciar los procesos distribuidos, mientras que los nodos esclavos permanecen a la espera de instrucciones para ejecutar las tareas asignadas.

- Tecnologías Utilizadas

- **Sistema operativo: Ubuntu (Linux)**, Seleccionado por su estabilidad, código abierto y compatibilidad nativa con entornos de computación distribuida y paralela.
- **Lenguaje de programación:** C/C++, utilizados para la implementación de los algoritmos paralelos, se optó por problemas en ese lenguaje debido a su alto rendimiento y compatibilidad con MPI.
- **SSH (Secure Shell):** Habilita el acceso remoto de forma segura y permite la administración eficiente de los nodos que conforman el clúster.
- **MPI (Message Passing Interface):** Framework estándar para la programación paralela y permite la comunicación y sincronización efectiva entre procesos distribuidos.
- **GCC (GNU Compiler Collection):** Conjunto de compiladores empleado para desarrollar y compilar aplicaciones en C/C++ dentro del entorno MPI.
- **Comandos de Red y Monitoreo del Sistema:** Se emplearon utilidades clave como ping para comprobar la conectividad entre los nodos, scp para la transferencia segura de archivos y htop/top para supervisar en tiempo real el uso de recursos del sistema, como: CPU, memoria y procesos.
- **Grafana:** Plataforma de código abierto que permite visualizar y ejecutar el análisis de las métricas de rendimiento del clúster.
- **Prometheus:** Conjunto de herramientas de código abierto para la monitorización del clúster, extrayendo datos de las aplicaciones, sistemas y servicios en ejecución, como: CPU y memoria, para su posterior visualización en Grafana.

- Configuraciones de RED y SSH

La configuración de red para el clúster, se estableció una red privada a través de una computadora dedicada como punto de acceso para la interconexión entre los nodos. El punto de acceso se utilizó una dirección IP (192.168.137.1), funcionando como puerta de enlace dentro de la red local.

Dentro de esta red se estableció un cluster compuesto por dos nodos:

- Nodo maestro: 192.168.137.3
- Nodo esclavo: 192.168.137.4

Ambos nodos se interconectan a través de un único punto de acceso y se establecieron dentro de la misma subred (192.168.137.0/24), lo que permitió una comunicación directa. Esta disposición garantiza la conectividad fundamental para el intercambio de datos y la ejecución remota de comandos, requisito clave para la operatividad de las tecnologías de la computación paralela.

Sobre esta infraestructura de red se implementó SSH (Secure Shell), configurando el acceso remoto seguro y sin contraseña entre el nodo maestro y el nodo esclavo, lo cual es indispensable para la ejecución de aplicaciones paralelas utilizando MPI y también para que el nodo maestro envíe los archivos al nodo esclavo.

El proceso de la configuración SSH sin contraseña es la siguiente:

- **Creación de la llave SSH en el nodo maestro:**

```
ssh-keygen -t rsa -b 4096
```

- **Copia de la clave pública al nodo esclavo:**

```
ssh-copy-id workeruser@192.168.137.4
```

Probar acceso sin contraseña:

```
ssh workeruser@192.168.137.4
```

- **Copia de la clave pública del esclavo al maestro:**

```
ssh-keygen -t rsa -b 4096
```

```
ssh-copy-id clusteradmin@192.168.137.3
```

Esto asegura que cualquier nodo pueda enviar y ejecutar programas en el otro.

Problemas encontrados

- Durante la preparación del entorno, se experimentó un fallo crítico de hardware en una de las computadoras destinadas a formar parte del cluster. El problema ocurrió durante el proceso de instalación de Ubuntu, resultando la eliminación del sistema operativo de Windows, la pérdida de varios documentos y la computadora dejó de detectar el disco duro principal de 1TB.
- El cluster configurado presenta una fuerte dependencia de la red en la que fue implementado inicialmente. Al intentar cambiar el punto de acceso en una computadora diferente a la asignada inicialmente, el clúster dejó de funcionar correctamente, perdiendo la comunicación entre los nodos maestro y esclavo.
- Durante la implementación del sistema de monitoreo del cluster, se presentaron dificultades significativas en la instalación de las dependencias necesarias para Prometheus y la configuración de su dashboard (panel) de visualización en Grafana.

Resultados

En la Fig. 6 se observa el despliegue del programa de multiplicación de matrices mediante la interfaz MPI, ejecutado con el comando mpirun. En esta etapa, el nodo maestro distribuye la carga de trabajo entre los nodos del clúster, permitiendo que múltiples procesos se ejecuten de forma simultánea. Este enfoque reduce el tiempo de cómputo en comparación con una ejecución secuencial, donde todas las operaciones se realizan en un solo procesador.

```

luis@ubuntu:~$ mpirun -np 12 --hostfile hosts ./matmul_mpi 1000
Authorization required, but no authorization protocol specified
MPI root: checksum diag = -336,504
luis@ubuntu:~$ 

```

Fig. 6 - Despliegue del programa matmul_mpi mediante la interfaz MPI.

En la Fig. 7 se observa la monitorización de los recursos del sistema mediante el comando top durante la ejecución del algoritmo. Se evidencia un uso elevado de la CPU por parte del proceso matmul_mpi, lo que confirma que el sistema se encuentra realizando cálculos intensivos de manera paralela. Este comportamiento indica un aprovechamiento eficiente de los recursos computacionales disponibles en el clúster.

PID	USUARIO	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	HORA+ ORDEN
7931	luis	20	0	4743824	34180	18440	R	100,3	0,2	0:14.39 matmul_mpi
7918	luis	20	0	4759092	49692	18776	R	100,0	0,3	0:14.41 matmul_mpi
7919	luis	20	0	4743832	34184	18440	R	100,0	0,2	0:14.42 matmul_mpi
7920	luis	20	0	4743836	34152	18408	R	100,0	0,2	0:14.42 matmul_mpi
7921	luis	20	0	4743892	34252	18509	R	100,0	0,2	0:14.41 matmul_mpi
7923	luis	20	0	4743824	34176	18436	R	100,0	0,2	0:14.41 matmul_mpi
7926	luis	20	0	4743824	34312	18572	R	100,0	0,2	0:14.41 matmul_mpi
7928	luis	20	0	4743888	34136	18396	R	100,0	0,2	0:14.42 matmul_mpi
2687	luis	20	0	4662748	361584	152648	S	11,6	2,3	1:53.16 gnome-shell
4463	luis	20	0	2175448	215792	139684	S	9,3	1,3	0:58.77 gnome-control-c
2381	luis	20	0	24,9g	181064	119192	S	8,3	1,1	1:18.99 Xorg
4118	root	20	0	0	0	0	I	2,7	0,0	0:08.44 kworker/u80:39-l915
2736	luis	20	0	1126972	119652	95304	S	1,0	0,7	0:06.68 mutter-x11-fran
1324	root	20	0	335156	21420	17388	S	0,7	0,1	0:02.54 NetworkManager
7534	root	0	-20	0	0	0	D	0,7	0,0	0:00.49 kworker/u810:0+l915_flip
18	root	20	0	0	0	0	I	0,3	0,0	0:02.64 rcu_preempt
2888	luis	20	0	541048	12644	18596	S	0,3	0,1	0:01.12 gsd-sharing
3106	luis	20	0	387020	8376	7272	S	0,3	0,1	0:00.00 gvfs-afc-volume
3772	luis	20	0	853240	67948	43436	S	0,3	0,4	1:53.59 gnome-terminal-
7730	luis	20	0	12236	6220	3998	R	0,3	0,0	0:00.69 top
7893	root	20	0	0	0	0	I	0,3	0,0	0:00.14 kworker/17:5-l915-unordered
1	root	20	0	23512	14680	9464	S	0,0	0,1	0:02.39 systend
2	root	20	0	0	0	0	S	0,0	0,0	0:00.01 kthreadread
3	root	20	0	0	0	0	S	0,0	0,0	0:00.00 pool_workqueue_release
4	root	0	-20	0	0	0	I	0,0	0,0	0:00.00 kworker/R-rcu_gp
5	root	0	-20	0	0	0	I	0,0	0,0	0:00.00 kworker/R-sync_wq
6	root	0	-20	0	0	0	I	0,0	0,0	0:00.00 kworker/R-kvfree_rcu_reclaim
7	root	0	-20	0	0	0	I	0,0	0,0	0:00.00 kworker/R-stub_flushwq
8	root	0	-20	0	0	0	I	0,0	0,0	0:00.00 kworker/R-neths
10	root	0	-20	0	0	0	I	0,0	0,0	0:00.00 kworker/B:0H-events_highpri
13	root	0	-20	0	0	0	I	0,0	0,0	0:00.00 kworker/R-mm_percpu_wq
14	root	20	0	0	0	0	I	0,0	0,0	0:00.00 rcu_tasks_kthread
15	root	20	0	0	0	0	I	0,0	0,0	0:00.00 rcu_tasks_rude_kthread
16	root	20	0	0	0	0	I	0,0	0,0	0:00.00 rcu_tasks_trace_kthread
17	root	20	0	0	0	0	S	0,0	0,0	0:00.05 ksfttfdq/0
19	root	20	0	0	0	0	S	0,0	0,0	0:00.00 rcu_exp_par_gp_kthread_worker/1

Fig. 7 - Monitorización de recursos mediante el comando top.

En la Fig. 8 se observan los resultados obtenidos en el nodo maestro tras la ejecución del algoritmo de multiplicación de matrices. El tiempo aproximado de resolución del problema en el clúster fue de 3.2 segundos, mientras que una ejecución secuencial del mismo algoritmo habría requerido alrededor de 9.5 segundos. Esto demuestra que el uso del clúster permite una reducción significativa del tiempo de ejecución gracias al paralelismo.

```

luis@ubuntu:~$ mpirun -np 12 --hostfile hosts ./matmul_mpi 1000
Authorization required, but no authorization protocol specified
MPI root: checksum diag = -336.504
luis@ubuntu:~$ 

```

Fig. 8 - Resultados obtenidos de la ejecución del clúster (nodo maestro).

En la Fig. 9 se observa la ejecución del algoritmo desde el nodo esclavo del clúster. Mediante la monitorización de procesos, se identifica la ejecución activa del programa matmul_mpi, lo que confirma que el nodo esclavo participa directamente en el procesamiento distribuido. Esta cooperación entre nodos es clave para disminuir el tiempo total de cómputo frente a un enfoque secuencial.

```

Descargas Escritorio Imágenes Música Pública sumaVector
Documentos hola matmul_mpi Plantillas snap Vídeos
bayron@bayron-ubuntu:~$ sudo apt-get install libopenmpi-dev
[sudo] contraseña para bayron:
Lo siento, prueba otra vez.
top - 10:50:52 up 3:16, 2 users, load average: 2,93, 2,23, 1,45
Tareas: 265 total, 6 ejecutar, 259 hibernar, 0 detener, 0 zombie
Cpu(s): 36,4 us, 62,7 sy, 0,0 nl, 0,0 id, 0,0 wa, 0,0 hl, 0,8 sl, 0,0 st
MiB Mem: 7638,7 total, 1556,6 libre, 3624,2 usado, 3471,0 búfcaché
MiB Intercambio: 4096,0 total, 4095,7 libre, 0,2 usado. 4014,5 dispon Mem

PID USUARIO PR NI VIRT RES SHR S %CPU %MEM HORA+ ORDEN
32250 bayron 20 0 227624 24732 12220 R 100,0 0,3 0:40.71 matmul_mpi
32249 bayron 20 0 226972 24072 12192 R 92,5 0,3 0:39.85 matmul_mpi
32248 bayron 20 0 226972 23980 12120 R 85,0 0,3 0:38.51 matmul_mpi
32247 bayron 20 0 227660 24648 12124 R 82,5 0,3 0:38.21 matmul_mpi
2867 bayron 20 0 4782168 342464 151772 S 12,1 4,4 7:06.30 gnome-shell
32304 bayron 20 0 784160 89240 46100 S 9,6 1,1 0:03.27 eog
517 root -51 0 0 0 0 0 5 6,7 0,0 2:06.50 irq/158-rtw88_pci
25868 bayron 20 0 629572 60712 47572 R 6,7 0,8 0:06.66 gnome-terminal-
284 root 19 -1 165772 97284 95644 S 1,7 1,2 0:46.45 systemd-journal
24667 bayron 20 0 1963496 221184 117796 S 0,8 2,8 1:56.50 gnome-control-c
39 root 20 0 0 0 0 5 0,4 0,0 0:01.44 ksoftirqd/3
138 root 0 -20 0 0 0 I 0,4 0,0 0:02.07 kworker/0:1H-i915_cleanup
183 root -51 0 0 0 0 5 0,4 0,0 0:10.16 irq/57-ELAN1200:00
831 message+ 20 0 12284 7732 4788 S 0,4 0,1 0:12.40 dbus-daemon
2201 bayron 20 0 386632 12864 7528 S 0,4 0,2 0:07.03 ibus-daemon
2391 bayron 20 0 419732 30864 19352 S 0,4 0,4 0:03.66 ibus-extension-
1566 bayron 20 10 1016702 34494 19474 S 0,4 0,4 0:08.99 tracker-miner-f

```

Fig. 9 - Resultados obtenidos de la ejecución del clúster (nodo esclavo).

En la Fig. 10 se observan las métricas de rendimiento del nodo maestro obtenidas a través de Grafana. Se aprecia un incremento en el uso de CPU y en la actividad de red durante la ejecución del algoritmo, lo cual evidencia la comunicación constante entre nodos y la correcta sincronización de los procesos paralelos.

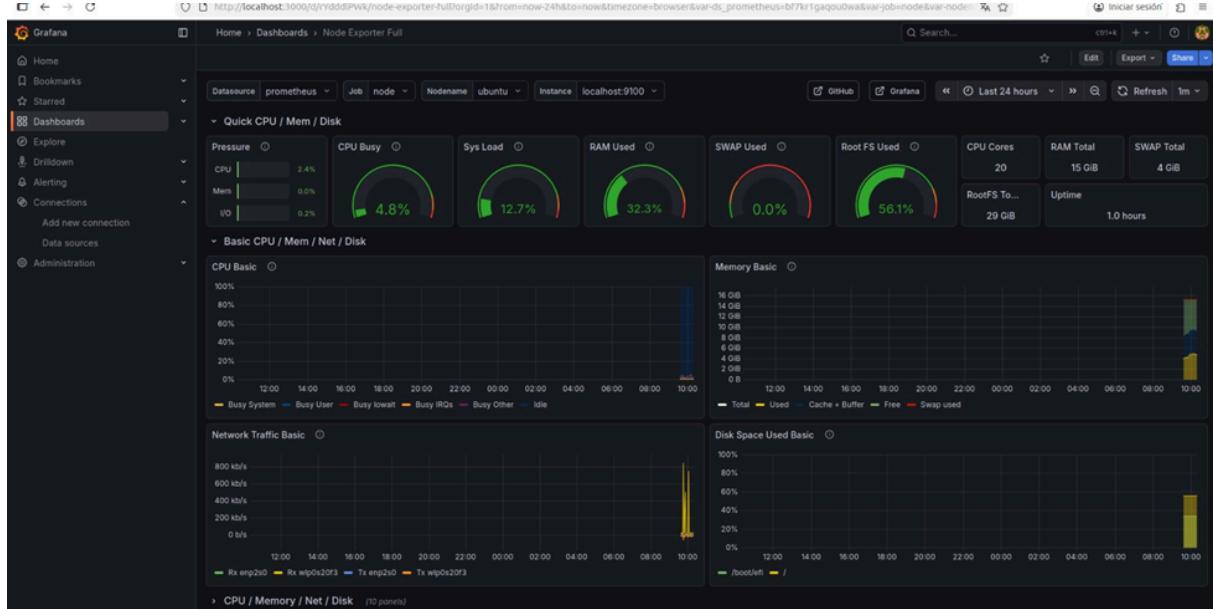


Fig. 10 - Métricas de rendimiento mediante la herramienta Grafana (nodo padre)

En la Fig. 11 se observan las métricas de rendimiento correspondientes al nodo esclavo. El comportamiento similar al del nodo maestro, con picos de uso de CPU durante la ejecución, confirma que ambos nodos trabajan de forma simultánea, reforzando la ventaja del clúster frente a la ejecución secuencial.

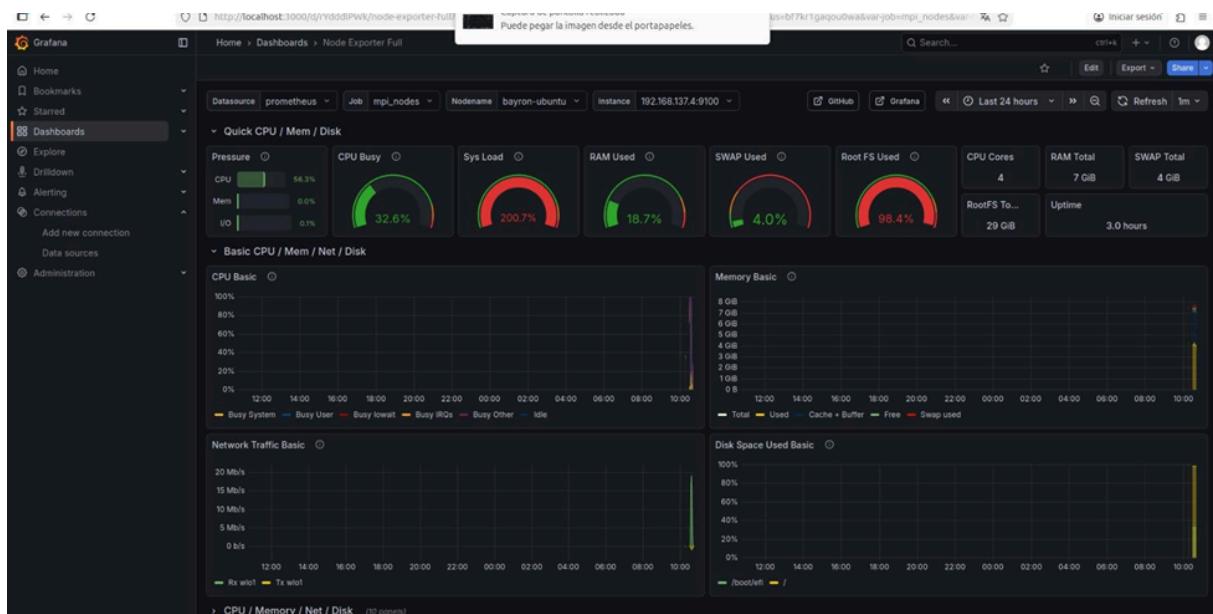


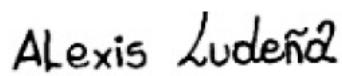
Fig. 11 - Métricas de rendimiento mediante la herramienta Grafana (nodo esclavo)

Gerardo Israel Herrera Campoverde

Bayron Alejandro Jimenez Cango



Nayely Cruzcaya Ramirez Herrera



Alexis Gradyn Ludeña Cueva



Luis Alberto Medina Chamba