USANDO UN CLUSTER

Matrices

Presentado por Leidy Johanna Martinez Ramirez y Valentín Valencia Valencia

Índice

1. Resumen

El presente informe aborda la implementación de la optimización en un sistema distribuido mediante un cluster con tres working nodes. Se usó una configuración NFS para que todas las máquinas tuvieran acceso a una carpeta compartida, esto es importante para la multiplicación de matrices porque A y B se guardan como archivos en ese folder y todas las máquinas deben leerlas.

2. Introducción

En el ámbito de la computación distribuida, la eficiencia y el rendimiento son aspectos críticos para garantizar un procesamiento rápido y efectivo de tareas complejas. Uno de los desafíos recurrentes en este dominio es la optimización de algoritmos para operaciones intensivas, como la multiplicación de matrices. Este informe se centra en el análisis y la mejora del rendimiento de un programa específico diseñado para calcular el tiempo requerido para multiplicar matrices de tamaño n x n en un entorno distribuido con un cluster basado en sistemas operativos Linux.

El programa en cuestión representa una aplicación crítica en numerosos campos, desde la simulación científica hasta el aprendizaje automático distribuido. El objetivo principal es maximizar la eficiencia de la multiplicación de matrices, teniendo en cuenta las complejidades inherentes a la distribución de tareas en un entorno de cluster.

3. Marco conceptual

3.1. Teoría de los algoritmos

3.1.1. Matrices

La multiplicación de matrices, aunque aparentemente sencilla, plantea desafíos significativos cuando se trata de matrices de gran tamaño. La elección de la estrategia de programación adecuada es crucial para lograr la máxima eficiencia computacional. Este informe se propone analizar y comparar detenidamente estos dos enfoques, con el objetivo de brindar una comprensión profunda de sus ventajas, desventajas y aplicaciones ideales.

3.1.2 Cluster en Linux

El cluster está compuesto por un conjunto de **tres working nodes** que trabajan de manera colaborativa para ejecutar tareas concurrentes, y el sistema operativo Linux proporciona el entorno necesario para coordinar estas operaciones distribuidas de manera eficiente.

Se usó una configuración NFS para que todas las máquinas tuvieran acceso a una carpeta compartida, esto es importante para la multiplicación de matrices porque A y B se guardan como archivos en ese folder y todas las máquinas deben leerlas.

3.2. Características de las máquinas

3.2.1 Máquina de Valentín

Característica	Especificación
Procesador	Procesador Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz, 1190 Mhz,
Sistema Operativo	ubuntu-22.04.3-live-server-amd64
Número de núcleos	4 procesadores principales, 8 procesadores lógicos

3.3. Desarrollo

3.3.1. Implementación del Cluster

- Configuración del Hardware:

  Se verificó la disponibilidad de hardware compatible para la construcción del cluster, asegurando la presencia de al menos dos nodos interconectados.

- Sistema Operativo:

  Se instaló una distribución de Linux en cada nodo, en este caso se hizo copiando una imagen de esta proporcionada por el docente.

- Conexión de Red:

  Se configuró la red para permitir la comunicación entre nodos, utilizando conexiones Ethernet o estableciendo una red privada según las necesidades.

- Configuración de SSH:

  Se implementó la autenticación SSH entre nodos para facilitar la administración remota y la ejecución de comandos.

- Instalación de Software de Cluster:

  Se seleccionó e instaló el software de gestión de clúster según los requisitos específicos de la aplicación, en este caso se usó OpenMP.

- Configuración del Software de Cluster:

  Se configuró el software de clúster para definir roles de nodos, establecer configuraciones de almacenamiento compartido y adaptar la infraestructura a las necesidades particulares del proyecto.

- Prueba de Comunicación:

  Se realizó una prueba exhaustiva para garantizar una comunicación fluida entre los nodos, empleando tanto comandos remotos como herramientas específicas del software de clúster utilizado.

- Desarrollo y Ejecución de Aplicaciones:

  Se desarrollaron aplicaciones específicas para aprovechar la capacidad de procesamiento distribuido del cluster, asegurando que estuvieran configuradas para ejecutarse eficientemente en este entorno. En este caso, se evaluó el rendimiento del programa para la multiplicación de matrices N x N.

4. Pruebas

4.1. Pruebas Valentin

Tamaño (N)	100	500	800	900	1000
1	0,014757	0,554282	3,155554	4,781991	7,394309
2	0,003828	0,659580	3,454550	4,362846	6,212509
3	0,004184	0,708793	2,793107	4,284970	6,724569
4	0,004727	0,730302	3,473376	4,420838	6,889278
5	0,005037	0,699917	3,042622	4,323811	5,963884
6	0,003428	0,650663	2,894060	4,277415	5,822118
7	0,004258	0,615590	2,908209	4,104324	6,159533
8	0,003581	0,653073	2,971956	4,221505	6,444328
9	0,003416	0,633322	3,106084	4,347475	6,510886
10	0,003429	0,668069	3,154142	4,245528	5,915940
Promedio	**0,005065**	**0,657359**	**3,095366**	**4,337070**	**6,403735**

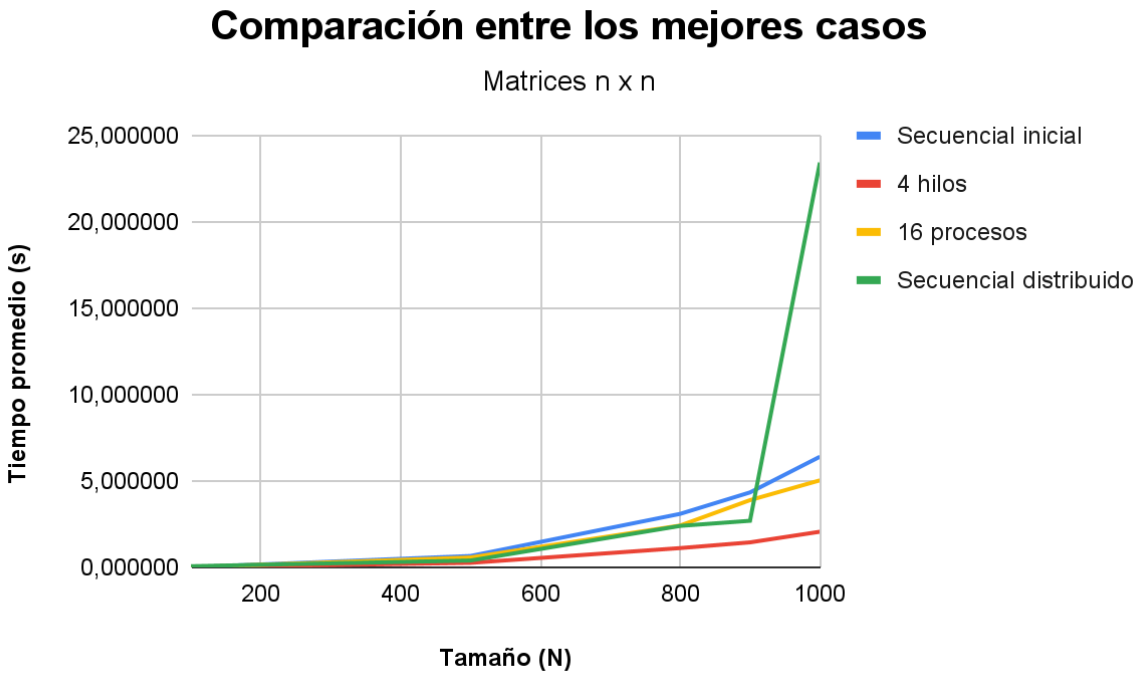
5. Resultados (Comparación)

5.1. Máquina Valentín

Es importante destacar, que se debió ajustar el tamaño de los N para su utilización en el sistema distribuido, puesto que curiosamente debieron ser múltiplos de tres. Este ajuste es solo de una cifra respecto a los tamaños anteriores.

Se tomaron los valores promedio de los tiempos en el secuencial del Caso de Estudio 01, y los valores de los cuatro hilos y dieciséis procesos del Caso de Estudio 02 para compararlos con los resultados de este Caso de Estudio 03.

Tamaño (N)	100	500	800	900	1000
Secuencial inicial	0,005065	0,657359	3,095366	4,337070	6,403735
4 hilos	0,002971	0,258636	1,114000	1,444773	2,061053
16 procesos	0,009120	0,572152	2,420348	3,889474	5,035520
Secuencial distribuido	0,065306	0,393728	2,397746	2,695132	23,435255



5. Conclusiones

- El mejor tiempo de ejecución, es decir, el más bajo, está dado por la programación por 4 hilos.
- El peor tiempo de ejecución (el más alto) está dado por la programación secuencial mediante un sistema distribuido.
- Sería interesante ver como se comportan los tiempos de ejecución del Caso de Estudio al configurar el cluster para que use toda la potencia del sistema distribuido y la distribución de cargas de trabajo mediante y los y/o procesos.

7. Bibliografía

[1] Cortéz, A. (2004). TEORÍA DE LA COMPLEJIDAD COMPUTACIONAL Y TEORÍA DE.

Revista De Investigación De Sistemas E Informática, 1(1), 102-105. Recuperado de: https://revistasinvestigacion.unmsm.edu.pe/index.php/sistem/article/view/3216

[2] Montero, L. H., & Antunez, R. R. (2011, 07). Parallel programming: definitions, mechanisms

and trouble. From

https://www.researchgate.net/publication/274960405_Parallel_programming_definitions_mechanisms_and_trouble

[3] Qaz Wiki. (2020). Qaz Wiki. From https://es.qaz.wiki/wiki/Matrix_multiplication_algorithm

[4] Arm. (2023). Arm Compiler for embedded User Guide. *developer.arm.com*. https://developer.arm.com/documentation/100748/0620/Using-Common-Compiler-Options/Selecting-optimization-options

[5] Wikipedia contributors. (2023). Inline expansion. *Wikipedia*. https://en.wikipedia.org/wiki/Inline_expansion

[6] *PI - unleashed*. (n.d.). Google Books. https://books.google.com.co/books?id=JlG5rFH7Ge0C&pg=PA39&lpg=PA39&dq=Dartboard+Method+algorithm&source=bl&ots=t76R30Q342&sig=NjguOYMc0ILqZs8Bcz6uIpfejdc&hl=en&ei=-YzTSuutFMefkQXj9_H7Aw&sa=X&oi=book_result&ct=result&redir_esc=y#v=onepage&q&f=false