

Proyecto Final

Curso de Sistemas Operativos y Laboratorio

Inversión de matrices de gran tamaño

Cristian Daniel Muñoz Botero
Jonathan Andrés Granda Orrego
Brandon Duque García
Jhon Alexander Botero Gomez

UdeA

2025

Resumen

Varios algoritmos de inteligencia artificial tales como máquinas de soporte vectorial, redes neuronales convolucionales entre otros, requieren dentro de sus procedimientos calcular matrices inversas que en muchos casos son matrices de gran tamaño. La inversión de una matriz es un problema que requiere $O[n^3]$ operaciones matemáticas, por lo que si la matriz es de gran tamaño puede demorarse demasiado su cálculo. Para reducir el tiempo del cálculo de la matriz inversa varios teoremas de álgebra lineal se han propuesto, buscando la estrategia "divide y vencerás". Un proyecto interesante antes de crear una propia aplicación de inteligencia artificial es elaborar un programa que implemente mediante procesos jerarquizados e hilos el cálculo de la inversa de una matriz de gran tamaño.

Introducción

- **¿Cuál es la necesidad /o problema que aborda el desafío seleccionado?**
 - El desafío aborda la necesidad de optimizar el cálculo de matrices inversas de gran tamaño, un proceso que es fundamental pero a la vez costoso computacionalmente en muchos algoritmos de inteligencia artificial. Dado que la inversión de matrices implica una complejidad computacional de orden $O[n^3]$, dicho cálculo puede convertirse en un cuello de botella en aplicaciones prácticas. Por ello se requiere una solución que permita reducir el tiempo de procesamiento mediante técnicas eficientes como una estructura jerárquica de procesos y paralelización con hilos.
- **¿Por qué es importante el desarrollo de este desafío en el contexto tecnológico actual?**
 - En el contexto tecnológico actual, donde el procesamiento eficiente de grandes cantidades de datos es fundamental para el desarrollo de aplicaciones de inteligencia artificial, optimizar operaciones algebraicas como la inversión de matrices resulta clave. La posibilidad de acelerar estos cálculos mediante paralelización de hilos contribuye significativamente al rendimiento de modelos de aprendizaje automático, permitiendo entrenar sistemas más complejos en menor tiempo. Además, el diseño de soluciones propias fortalece la capacidad de innovación y comprensión de las bases matemáticas que sustentan la inteligencia artificial moderna.

Antecedentes o marco teórico

- **¿Cuáles son los principales aspectos teóricos necesarios para comprender el desafío y llevarlo a cabo?**

- Para comprender y desarrollar este desafío es necesario dominar varios aspectos teóricos clave. En primer lugar, se requiere conocimiento previo de álgebra lineal, específicamente en lo relacionado con matrices, operaciones matriciales e inversión de matrices, así como teoremas que permitan su descomposición o simplificación, un ejemplo es descomposición por LU.
- El uso e implementación de cálculos algoritmos mediante el uso de GPUs haciendo uso de algoritmos y métodos más comunes como Factorización LU o eliminación Gaussiana aprovechando arquitecturas de GPUs, tal y como lo es CUDA (Compute Unified Device Architecture).

- **¿Qué relación tiene esta teoría con los temas del curso de Sistemas Operativos (en la parte teórica y para el laboratorio)?**

Este desafío se relaciona directamente con varios temas fundamentales del curso, tanto en su componente teórico como práctico.

- Como primero, debemos saber que invertir matrices de gran tamaño implica almacenar una cantidad masiva de datos, se debe gestionar eficientemente la memoria RAM disponible.
- También se hace de bibliotecas optimizadas, que utilizan bibliotecas numéricas que suelen estar optimizadas en las arquitecturas, haciendo uso intensivo de los recursos
- También involucran conceptos como gestión de procesos e hilos, ya que el desafío requiere ejecutar múltiples hilos o procesos para distribuir el trabajo del cálculo de la matriz inversa.
- Dado el mismo algoritmo de inversión de las matrices que se usará se pueden implementar algoritmos de computación paralela para reducir el tiempo total que consumen dichos algoritmos.

Objetivos (principal y específicos)

Objetivo general

Desarrollar algoritmos y realizar comparaciones en cuanto a eficiencia y escalabilidad para la inversión de matrices de gran tamaño, que hagan uso eficiente de los recursos de hardware

Objetivos específicos

1. Seleccionar y analizar distintos algoritmos de inversión de matrices, tales como lo son descomposición LU eliminación Gauss-Jordan o métodos iterativos.

2. Plantear y evaluar estrategias de paralelización de algoritmos en diferentes lenguajes como C y python, implicando partes del algoritmo que pueden ejecutarse de manera aislada e independiente determinando la manera adecuada de distribuir el trabajo
3. Implementar el algoritmo paralelizado utilizando herramientas y APIS del sistema operativo, involucrando directamente el uso de bibliotecas de hilos (como **pthread** en sistemas POSIX o el módulo **threading** en python)
4. Evaluar el rendimiento de las implementaciones en diferentes tamaños de matriz, números de hilos, haciendo pruebas al medir tiempos de ejecución, utilización de CPUs y uso de la memoria
5. Probar el funcionamiento de los algoritmos mediante ejecuciones con arquitecturas basadas en GPUs

Metodología

- **¿Cuáles son las principales herramientas que podrían utilizarse para implementar la solución?**
 - Para implementar la solución propuesta, se pueden utilizar varias herramientas tanto a nivel de software como de programación. En primer lugar, el lenguaje C es una opción sólida debido a su eficiencia y control sobre recursos del sistema, especialmente al trabajar con procesos, hilos y manejo de memoria dinámica. Al nivel de sistema operativo, el entorno Linux ofrece las herramientas y bibliotecas necesarias para la gestión de procesos e hilos, así como comandos y utilidades para la supervisión del rendimiento. Además de que podemos usar herramientas de depuración y análisis como Valgrind y GDB que facilitan la detección de errores y el control de uso de recursos, principalmente de memoria.
- **¿Qué actividades son necesarias para cumplir los objetivos? (Tenga en cuenta el orden lógico para el desarrollo de las actividades).**

Para cumplir los objetivos consideramos necesario seguir el siguiente orden:

- **Revisión teórica y documental:**
 - Investigación sobre inversión de matrices, métodos numéricos eficientes como descomposición por LU y fundamentos de programación paralela y concurrente.
- **Diseño de algoritmo general:**
 - Plantear la estrategia que se usará para dividir el cálculo de la matriz inversa en tareas paralelas considerando cómo se dividirá la matriz y cómo se unirán los resultados.
- **Selección de herramientas de entorno de desarrollo:**
 - Definir el lenguaje de programación, sistema operativo y bibliotecas que se emplearán.
- **Implementación de una versión piloto:**

- Crear una primera versión del programa que calcule la inversa de una matriz sin paralelismo en lenguaje C y Python, que nos sirva como base funcional.
- **Implementación del módulo paralelo**
 - Implementar la versión paralela del cálculo, utilizando procesos e hilos para dividir las tareas.
- **Comparación de tiempos:**
 - Comparar el tiempo obtenido en todas las implementaciones.
- **Conclusiones del estudio:**
 - Presentar las principales conclusiones derivadas del análisis realizado previamente.

Cronograma

	18 mayo - 24 mayo	25 mayo - 31 mayo	1 junio - 7 junio	8 junio - 14 junio	15 junio - 21 junio	22 junio - 28 de junio	29 junio - 5 julio
Revisión teórica y documental							
Diseño de los algoritmos en pseudocódigo							
Implementación del algoritmo sin paralelismo en C							
Implementación del algoritmo sin paralelismo en Python							
Implementación del algoritmo con paralelismo en C							
Implementación del algoritmo con paralelismo en Python							
Comparación de tiempos de todas las implementaciones							
Conclusiones del estudio							

Referencias

De ecuaciones lineales, S. S., Lineales, de E., & De matrices y mínimos cuadrados, I. (s/f). Algoritmos matemáticos para: Ula.ve. Recuperado el 13 de mayo de 2025, de <https://www.ing.ula.ve/~aguilar/actividad-docente/AYDA/Clase8MiniSem.pdf>

Ezzatti, P., Quintana-Ortí, E. S., & Remón, A. (s/f). Uso de GPUs para acelerar el cálculo de la matriz inversa. Clei.org. Recuperado el 13 de mayo de 2025, de https://clei.org/proceedings_data/CLEI2010/CLEI2010/11_ArquitecturaDeComputadoras/4.2.2_43CLEIEzzatti_Paper.pdf