UNIVERSIDAD DE ANTIOQUIA
1803
Escuela Interamericana de Bibliotecología

Inversión de matrices de gran tamaño

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

UNIVERSIDAD DE ANTIOQUIA
Escuela Interamericana de Bibliotecología

Introducción

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.

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

- Analizar algoritmos de inversión de matrices (LU, Gauss-Jordan, métodos iterativos).
- Diseñar estrategias de paralelización en C y Python, identificando tareas independientes.
- Implementar los algoritmos usando APIs y bibliotecas de hilos (pthreads, threading).
- Evaluar el rendimiento con diferentes tamaños de matriz, cantidad de hilos y recursos usados.
- Probar la ejecución en arquitecturas con soporte para GPU.

UNIVERSIDAD DE ANTIOQUIA
Escuela Interamericana de Bibliotecología

Inversión de matrices de gran tamaño

Importancia
Optimizar la inversión de matrices es vital en simulación, gráficos y machine learning.

Aplicaciones
Es importante tener claro que estas operaciones computacionales, tienen aplicación en modelos físicos hasta la inteligencia artificial

Algoritmos clásicos de Inversión de matrices

UNIVERSIDAD DE ANTIOQUIA
Escuela Interamericana de Bibliotecología

Eliminación Gaussiana
Suele ser un algoritmo simple a la hora de interpretar numéricamente, pero resulta bastante costoso a la hora de aumentar las dimensiones de la matriz.

Factorización o Descomposición LU
Este algoritmo puede presentar variaciones y optimizaciones que presentan más eficiencia con respecto a otros

Demas metodos…

- **Cholesky,** se limita a matrices simétricamente positivas
- **QR,** una alternativa más estable de manera numérica

Metodología – Herramientas y Entorno

- **Lenguaje C:** Alta eficiencia y control de recursos,ideal para manejar procesos e hilos.
- **Python:** Complemento útil para prototipos y comparación de resultados.
- **Linux:** Entorno robusto con herramientas para gestión de procesos e hilos.
- **Bibliotecas y utilidades:**
  - Pthreads: Hilos en c
  - Threading: Python
  - Valgrind y GDB: Depuración y análisis de memoria

Metodología – Actividades del Proyecto

- **Revisión teórica:** Estudio de métodos de inversión de matrices y programación paralela.
- **Diseño de algoritmo:** Estrategia para dividir y combinar la matriz en tareas paralelas.
- **Selección de herramientas:** Definición del lenguaje, SO y bibliotecas.
- **Implementación base:** Versión secuencial funcional en C y python.
- **Paralelización:** Desarrollo de la versión paralela usando procesos e hilos.
- **Evaluación de rendimiento:** Comparación de tiempos y análisis de recursos.
- **Conclusiones:** Resultados obtenidos y aprendizajes clave del proyecto.

Implementación por lenguajes. Ventajas y desventajas

- **Herramientas o Bibliotecas:** Numpy y SciPy facilitan el trabajo y la representación de matrices.
- **Limitaciones:** GIL reduce la eficacia de hilos,es un mecanismo interno de control que garantiza que solo un hilo de ejecución acceda a objetos en un momento dado.
- **Alternativas:** Multiprocessing y Cython para mejorar el rendimiento, junto con Python Threads

Implementación por lenguajes. Ventajas y desventajas

- **Herramientas o Bibliotecas:** BLAS y LAPACK, incluyendo Intel MKL y OpenBLAS para rendimiento.
- **Limitaciones:** Representaciones de estructuras, crear módulos o componentes de código
- **Alternativas:** OpenMP brinda un excelente manejo de hilos para la aceleración de cálculos

UNIVERSIDAD DE ANTIOQUIA
Escuela Interamericana de Bibliotecología

Consideraciones y aspectos preliminares

Tiempos de ejecución
El lenguaje C puede ofrecer tiempos con mejor rendimiento en el momento que las dimensiones de matrices empiecen a aumentar

Uso de CPU
C ayuda a maximizar el uso de aplicaciones e implementaciones multihilo

Escalabilidad
La herramienta Cython ofrece un balance de rendimiento y facilidad a la hora de implementar soluciones

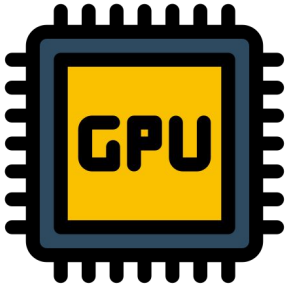
Funcionamiento de procesos jerarquizados

1. **División**
   La matriz se divide en submatrices para procesar en paralelo.
2. **Comunicación**
   MPI(Interfaz de paso de mensajes) sincroniza los procesos distribuidos entre nodos, ya que podemos comunicar procesadores en un cluster
3. **Estrategias**
   Distribuir por filas, columnas o bloques optimiza carga.

UNIVERSIDAD DE ANTIOQUIA
Escuela Interamericana de Bibliotecología

Posibles alternativas o retos

Arquitecturas
Uso de GPUs con **CUDA** y **OpenCL** ya que son modelos que prometen gran aceleración en los procesos de cómputo



Algoritmos iterativos
Algoritmos optimizados para sistemas dispersos como **gradiente conjugado**, basados en **Krylov, GMRES, BiCGSTAB**

La computación cuántica permite resolver sistemas lineales en tiempos logarítmicos de manera teórica, aunque aún es experimental promete una buena línea.

UNIVERSIDAD DE ANTIOQUIA
Escuela Interamericana de Bibliotecología

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