PHÂN TÍCH VÀ SO SÁNH HIỆU NĂNG XỬ LÝ TOÁN HỌC MA TRẬN TRÊN KIẾN TRÚC CPU VÀ GPU

Tác giả: Hoàng Gia Hùng
Faculty of Computer Engineering
University of Information Technology UIT-VNUHCM
Contact: hoang.ghung.forwork@gmail.com
25520624@gm.uit.edu.vn

Tóm tắt: Ma trận là cấu trúc dữ liệu nền tảng trong nhiều lĩnh vực khoa học máy tính, đặc biệt là Trí tuệ nhân tạo (AI) và Mô phỏng thực nghiệm. Việc lựa chọn phần cứng xử lý (CPU hay GPU) đóng vai trò quyết định đến hiệu suất của các thuật toán dựa trên ma trận. Nghiên cứu này thực hiện phân tích và so sánh hiệu năng thực thi của hai phép toán cơ bản: nhân ma trận và tìm ma trận nghịch đảo trên hai nền tảng kiến trúc khác biệt. Kết quả thực nghiệm cho thấy sự vượt trội của GPU đối với các ma trận kích thước lớn nhờ khả năng tính toán song song, trong khi CPU vẫn duy trì lợi thế ở các bài toán quy mô nhỏ do độ trễ thấp. Từ các số liệu về thời gian thực thi và ước tính năng lượng, bài báo đưa ra các đánh giá định lượng giúp tối ưu hóa việc lựa chọn tài nguyên tính toán cho các ứng dụng cụ thể.

I. Giới thiệu

1.1 Kiến trúc và đặc thù xử lí

Kiến trúc CPU truyền thống được thiết kế tối ưu cho việc xử lý tuần tự (Serial Processing) với mục tiêu giảm thiểu độ trễ Với số lượng lõi ít nhưng xung nhịp cao và bộ dự đoán rẽ nhánh phức tạp, CPU xử lý hiệu quả các tác vụ logic đa dạng, điều phối tài nguyên và quản lý hệ điều hành [1].

Trái ngược lại, GPU (Graphics Processing Unit) được thiết kế để khắc phục các hạn chế về băng thông tính toán của CPU. Với kiến trúc hàng nghìn lõi nhỏ (Ví dụ: kiến trúc NVIDIA trên RTX 4090 sở hữu hơn 16.000 lõi CUDA [2]), GPU ưu tiên thông lượng (Throughput) và khả năng xử lý song song (Parallel Computing). Đặc điểm này khiến GPU trở nên lý tưởng cho các tác vụ SIMD (Single Instruction Multiple Data) như tính toán ma trận, nơi cùng một phép toán được áp dụng lên khối lượng dữ liệu lớn. Thay vì xử lý tuần tự từng phần tử, GPU chia nhỏ bài toán thành các khối (Blocks) và luồng (Threads) để thực thi đồng thời.

1.2 Vấn đề nghiên cứu

Tuy nhiên, liệu GPU có luôn vượt trội hơn CPU trong mọi kịch bản? Việc phân tách dữ liệu và truyền tải giữa RAM (Host) và VRAM (Device) có tạo ra độ trễ làm giảm hiệu năng ở các bài toán nhỏ? Nghiên cứu này sẽ tập trung trả lời các câu hỏi trên thông qua thực nghiệm với phép nhân ma trận và tính ma trận nghịch đảo, từ đó cung cấp cái nhìn trực quan về "điểm giao cắt" hiệu năng giữa hai kiến trúc.

1.Mục tiêu nghiên cứu:

1.1 So sánh thời gian thực thi (Excecution time) của CPU và GPU trên phép nhân ma trận.

1.2 Xác định từ kịch thước vào GPU vượt trội hơn hoặc ngược lại, phân tích đến những yếu tố có thể ảnh hưởng đến hiệu năng: đường truyền dữ liệu và thư viện tối ưu.

2. Phương pháp nghiên cứu:

2.1: Môi trường thực nghiệm thực hiện trên Google Collab.
Phần cứng: GPU: Tesla T4.
Phần mềm và thư viện: Thư viện CPU: Numpy 2.0.2 , Thư viện GPU: Cupy 13.6.0, Python 3.12.12.

2.2: Phương pháp đo lường và ghi nhận kết quả:

2.1.Khởi tạo ma trận ngẫu nhiên với kích thước N

2.2 Chạy khởi động (Warm-up) để loại bỏ thời gian khởi tạo context của CUDA ban đầu.

2.3. Đo thời gian thực thi bằng hàm time.perf_counter(). Lệnh cupy.cuda.Device().synchronize() trước và sau khi đo để đảm bảo tính toán đã hoàn tất.

2.4.Kết quả được lấy trung bình cộng sau 10 lần chạy thử nghiệm cho mỗi kích thước N.

3. Ý nghĩa và đóng góp:

Đưa ra những nhận định trực quan về CPU và GPU, từ đó làm nền tảng cho việc chọn dữ liệu phù hợp, đồng thời giúp cho duy thiết kế thuật toán trở nên tối ưu hơn,

[1] Hennessy, J. L., & Patterson, D. A. (2017). Computer Architecture: A Quantitative Approach.
[2] NVIDIA. (2022). NVIDIA Ada Lovelace Architecture Whitepaper

ngoài ra còn là tài liệu tham khảo về CPU và GPU giải thuật toán từ đơn giản đến phức tạp.

II. Nội dung

1. Thiết lập thí nghiệm:

1.1 Môi trường thí nghiệm:

Phần cứng: GPU: Tesla T4, VRAM: 14.74 GB

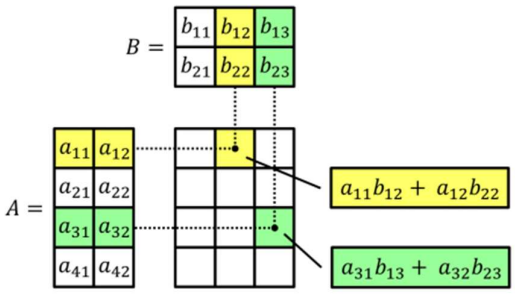
Phần mềm: Python, Numpy (CPU backend), Cupy (CUDA backend).

1.2 Nguyên tắc kiểm thử:

Để đảm bảo tính chính xác và khách quan dữ liệu đo lường, quá trình thiết lập thí nghiệm tuân thủ các nguyên tắc sau của lập trình hiệu năng cao (HPC): Bao gồm: Khởi động (Warm-up), đảm bảo thực hiện tính toán nháp để khởi động ngữ cảnh CUDA (CUDA Context) và nạp thư viện vào bộ nhớ Cache, loại bỏ độ trễ ban đầu (Latency), đồng bộ hóa (Synchroniztion) để đảm bảo thời gian ghi nhận thời gian thực tế, kết quả phải được lấy mẫu trung bình (Average Excution Time).

1.3 Phép toán khảo sát:

Phép nhân ma trận:



Phép nghịch đảo ma trận:

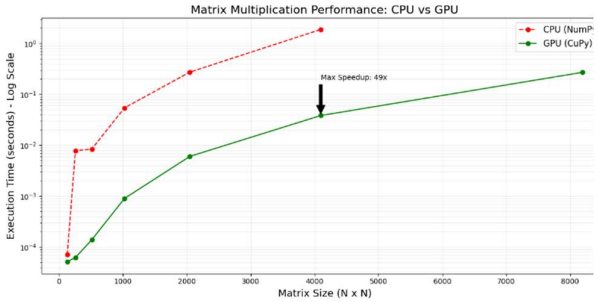
$$A^{-1} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}^{-1} = \frac{1}{|A|}\begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \end{bmatrix}^{T}$$

$$= \frac{1}{|A|}\begin{bmatrix} A & D & G \\ B & E & H \\ C & F & I \end{bmatrix}$$

$$|A| = a(ei - fh) - b(id - fg) + c(dh - eg)$$

$$A = (ei - fh) \quad D = -(bi - ch) \quad G = (bf - ce)$$
$$B = -(di - fg) \quad E = (ai - cg) \quad H = -(af - cd)$$
$$C = (dh - eg) \quad F = -(ah - bg) \quad I = (ae - bd)$$

2. Kết quả thí nghiệm



Hình 1: Biểu đồ so sánh thời gian thực thi nhân ma trận trên CPU và GPU theo kích thước N.

Kích thước N	CPU time (s)	GPU time (s)	Tốc độ
128	0.00007	0.00005	1.40
256	0.007512	0.00006	125.2
512	0.00841	0.00014	60.78
1024	0.05425	0.00090	60.33
2048	0.27419	0.00604	45.92
4096	1.88687	0.03833	49.20
8192	>4.0000	0.27067	N/A

Bảng 1: Bảng dữ liệu thời gian thực thi ma trận kích thước N và tỷ lệ tốc độ giữa CPU va GPU

Quan sát biểu đồ (Hình 1) và bảng dữ liệu chi tiết, có thể chia hệ thống thành 3 giai đoạn rõ rệt dựa trên kích thước của ma trận N.

Giai đoạn1: Tại N=128. Thời gian xử lí của cả hai kiến trúc đều nằm ở mức dưới 1ms, tuy nhiên tại N=256, độ chênh lệch của GPU và CPU đạt sự đột biến về tốc độ ~90 lần so với N=128, ở kích thước N=256, dữ liệu vẫn đủ nhỏ để nằm gọn trong Cache L1/L2 của CPU nhưng viêc xử lí tuần tự đã bắt đầu gặp trở ngại khi số lượng phép tính tăng lên, trong khi đó, với GPU, thời gian 0.00006s thực chất là thời gian gọi kernel (Kernel launch overhead) chứ không phải thời gian thực sự, mặc dù thời gian này gần như không đổi so với khi thực hiện với N=128 nhưng tỷ lệ chênh lệch với CPU tăng vọt.

Giai đoạn 2: Vùng ổn định hiệu năng (512<N<4096). Khi ma trận tăng dần, chỉ số tăng tốc (Speed run) bắt đầu đi vào quỹ đạo ổn định, giao động trong khoảng 45x-60x. Đây là giai đoạn phản ánh trung thực về sức mạnh của hai kiến trúc khi xử lí dữ liệu thô (Raw material). Đường biểu diễn của CPU (Red line) trên thang đo Logarit có độ dốc lớn hơn, cho thấy thời gian tăng nhanh theo cấp số nhân nhanh chóng, ngược lại đường biểu diễn GPU (Blue line) có xu hướng thoải, chứng tỏ khả năng hoạt đông tốt hơn nhiều trước sự phức tạp của thuật toán.

Giai đoạn 3: Mở rộng tính toán (N>8192) . Thời gian thực thi của CPU đa vượt qua ngưỡng (4s) thoe xu hướng dự đoán (Predictive trend) để xử lí, trong khi đó GPU vẫn hoạt động ổn định với thời gian hoàn thành 0.27s. Kết quả cho thấy đối với các bài tón Deep Learning hiện đại (thường làm việc với các tensor hàng chục nghìn) thì việc dùng GPU là yêu cầu bắt buộc.