ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

VIỆN TOÁN ỨNG DỤNG VÀ TIN HỌC
School of Applied Mathematics and Informatics

Tính toán song song

Chủ đề:
Chương trình tính tích vô hướng của 2 vector

Giảng viên giảng dạy: TS. Vũ Thành Nam
Mã học phần: MI4364
Mã lớp học: 146169
Nhóm sinh viên thực hiện: Nhóm 27

Vũ Văn Nghĩa	20206205
Trần Minh Quang	20206209
Nguyễn Quốc Thái	20206168

Ngày 28 tháng 11 năm 2023

Mục lục

Bảng đánh giá thành viên

BẢNG ĐÁNH GIÁ THÀNH VIÊN		
STT	**Họ tên**	**Nhiệm vụ**
1	Vũ Văn Nghĩa	Làm báo cáo LaTeX Lập trình tuần tự tính tích vô hướng 2 vector
2	Trần Minh Quang	Lập trình tuần tự nhân ma trận với vector Lập trình song song nhân ma trận với vector
3	Nguyễn Quốc Thái	Lập trình song song tính tích vô hướng 2 vector

Danh sách bảng

Danh sách hình vẽ

Danh sách mã nguồn

Lời mở đầu

Trong thời đại hiện nay, sự phát triển nhanh chóng của công nghệ đang thách thức với việc xử lý công việc ngày càng lớn và phức tạp. Tính toán song song là một phương pháp quan trọng để tăng cường hiệu suất tính toán, trong đó nhiều nhiệm vụ được thực hiện đồng thời, giúp tăng cường khả năng xử lý và giảm thời gian cần thiết để hoàn thành công việc.

Nhận thấy tầm quan trọng của tính toán song song giúp đáp ứng với thách thức ngày càng tăng của các vấn đề lớn và phức tạp trong nhiều lĩnh vực khác nhau. Nhóm 27 chúng em xin phép trình bài về chủ đề *"Chương trình tính tích vô hướng của 2 vector"* có sử dụng OpenMP để thực hiện tính toán song song. Thông qua báo cáo này, nhóm em hy vọng có thể đưa ra những giải pháp hiệu quả và áp dụng vào các vấn đề thực tế.

Nội dung của bài báo cáo bao gồm các nội dung sau:

- Thực hiện phép nhân ma trận với vector.
- Thực hiện chương trình tính tích vô hướng của 2 vector.

Nhóm chúng em xin gửi lời cảm ơn đến thầy TS. Vũ Thành Nam, người đã trực tiếp hướng dẫn, giảng dạy và truyền tải cho chúng em những kiến thức bổ ích của học phần. Trong quá trình hoàn thành báo cáo, mặc dù chúng em đã cố gắng tìm hiểu và hoàn thiện nhưng không thể tránh khỏi những thiếu sót, vậy nên chúng em rất mong nhận được những ý kiến nhận xét và góp ý của thầy và các bạn để báo cáo của nhóm được hoàn thiện hơn.

Chúng em xin chân thành cảm ơn!

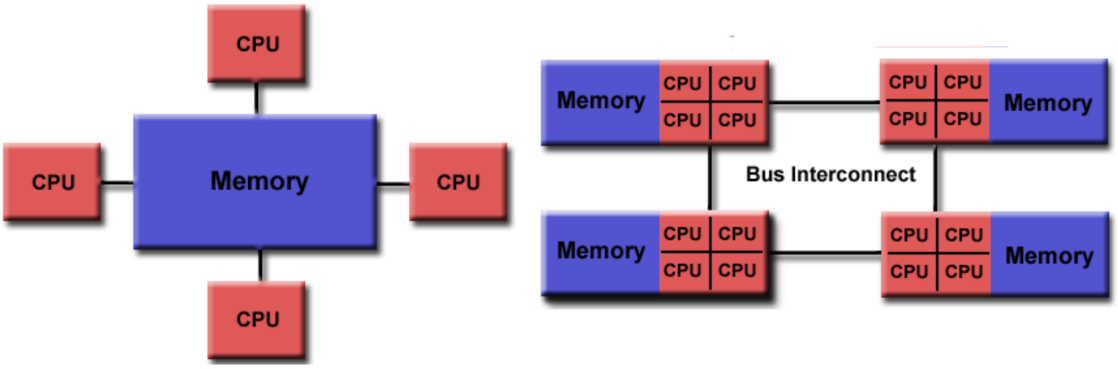
1 Giới thiệu chung

1.1 Giới thiệu chung về OpenMP

Lập trình đa luồng là một phương pháp lập trình trong đó một chương trình có khả năng thực hiện nhiều công việc đồng thời, mỗi công việc được thực hiện trong một luồng riêng biệt.

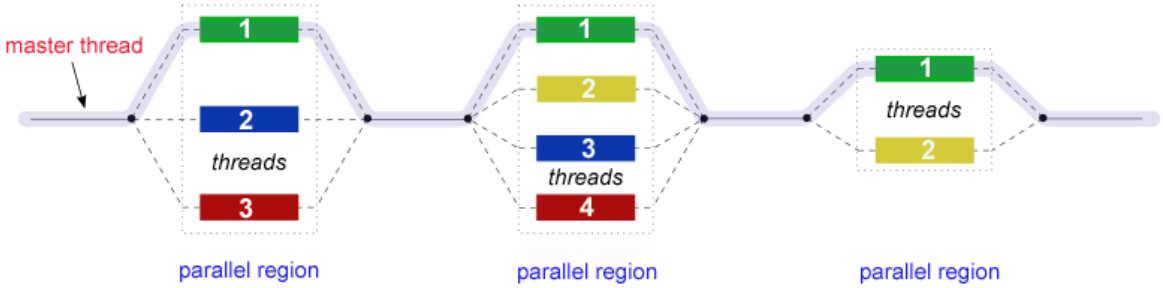
OpenMP (Open Multi-Processing) là một giao diện lập trình ứng dụng hỗ trợ lập trình đa xử lý bộ nhớ dùng chung đa nền tảng trong C , C++ và Fortran.

OpenMP được thiết kế cho các máy đa bộ xử lý, bộ nhớ dùng chung. Kiến trúc cơ bản có thể là bộ nhớ chia sẻ UMA hoặc NUMA. Chia sẻ bộ nhớ trong đó nhiều luồng có thể truy cập cùng một không gian bộ nhớ chung.



Hình 1: ViDuHinhAnhTheoChieuNgang

OpenMP sử dụng mô hình fork-join để thực thi song song:



Hình 2: ViDuHinhAnhTheoChieuNgang

Tất cả các chương trình OpenMP đều bắt đầu dưới dạng một tiến trình duy nhất: luồng chính . Luồng chính thực thi tuần tự cho đến khi gặp cấu trúc vùng song song đầu tiên.

FORK: luồng chính sau đó tạo ra một nhóm các luồng song song .

Sau đó, các câu lệnh trong chương trình được bao quanh bởi cấu trúc vùng song song sẽ được thực thi song song giữa các luồng nhóm khác nhau.

JOIN: Khi các luồng của nhóm hoàn thành các câu lệnh trong cấu trúc vùng song song, chúng sẽ đồng bộ hóa và chấm dứt, chỉ để lại luồng chính.

1.2 Công thức tính hiệu suất

Công thức tính hiệu suất trong bài báo cáo:

$$\text{Hiệu suất} = \frac{\text{Thời gian tuần tự}}{\text{Thời gian song song} \times \text{Số luồng}}$$

1.3 Thông tin về máy tính và phần mềm

Thông Tin CPU	Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz
Thông Tin RAM	16.0 GB
Hệ Điều Hành	Windows 11
Ngôn Ngữ Lập Trình	C, C++
Trình Biên Dịch	g++

Bảng 1: Thông Tin Hệ Thống

2 Thực hành chung

2.1 Yêu cầu

Yêu cầu

Thực hiện phép nhân ma trận với vector.
INPUT:

- Ma trận có m hàng và n cột.
- Vector có n phần tử.

OUTPUT:

- Vector kết quả của phép nhân ma trận với vector.

2.2 Cơ sở toán học

Phép nhân ma trận với vector là một phép toán phổ biến trong đại số tuyến tính. Công thức tính toán như sau:

$$\mathbf{w} = A \cdot \mathbf{v}$$

Trong đó:

- Với A là ma trận có kích thước $m \times n$.
- Với $\mathbf{v}$ là vector có kích thước $n \times 1$.
- Với $\mathbf{w}$ là vector kết quả có kích thước $m \times 1$.

Để tính giá trị của $\mathbf{w}$, thực hiện các phép nhân và cộng tương ứng:

$$\mathbf{w}_i = \sum_{j=1}^{n} A_{ij} \cdot v_j$$

Trong đó:

- Với A_{ij} là phần tử ở hàng i, cột j của ma trận A.
- Với v_j là phần tử thứ j của vector $\mathbf{v}$.
- Với $\mathbf{w}_i$ là phần tử thứ i của vector kết quả $\mathbf{w}$.

Ví dụ: Cho ma trận A và vector $\mathbf{v}$ sau:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix}$$

Phép nhân ma trận A với vector $\mathbf{v}$ sẽ tạo ra vector $\mathbf{w}$ như sau:

$$\mathbf{w} = A \cdot \mathbf{v}$$

Công thức tính toán:

$$\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix}$$

Thực hiện phép nhân và cộng:

$$w_1 = 1 \cdot 2 + 2 \cdot 1 + 3 \cdot 3 = 13$$

$$w_2 = 4 \cdot 2 + 5 \cdot 1 + 6 \cdot 3 = 31$$

Do đó, vector kết quả $\mathbf{w}$ là:

$$\mathbf{w} = \begin{bmatrix} 13 \\ 31 \end{bmatrix}$$

2.3 Mã nguồn phép nhân ma trận với vector

2.3.1 Mã nguồn lập trình tuần tự phép nhân ma trận với vector

```cpp
#include <iostream>

int main() {
    std::cout << "Hello from hello.cpp" << std::endl;
    return 0;
}
```

Mã nguồn 1: Mã nguồn lập trình tuần tự phép nhân ma trận với vector

2.3.2 Mã nguồn lập trình song song phép nhân ma trận với vector

```cpp
#include <iostream>

int main() {
    std::cout << "Hello from hello.cpp" << std::endl;
    return 0;
}
```

Mã nguồn 2: Mã nguồn lập trình song song phép nhân ma trận với vector

2.4 Kết quả:

2.4.1 Bảng kết quả phép nhân ma trận với vector

BẢNG KẾT QUẢ													
Kích thước	Thời gian tuần tự	Song song 2 luồng		Song song 3 luồng		Song song 4 luồng		Song song 5 luồng		Song song 6 luồng		Song song 7 luồng	
		Thời gian	Hiệu suất	Thời gian	Hiệu suất	Thời gian	Hiệu suất	Thời gian	Hiệu suất	Thời gian	Hiệu suất	Thời gian	Hiệu suất
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113

Bảng 2: Bảng Kết Quả

2.4.2 Biểu đồ phép nhân ma trận với vector

```
Successfully generated 2 random vectors of length: 10
Successfully generated 2 random vectors of length: 100
Successfully generated 2 random vectors of length: 1000
Successfully generated 2 random vectors of length: 10000
Successfully generated 2 random vectors of length: 100000
Successfully generated 2 random vectors of length: 1000000
Successfully generated 2 random vectors of length: 10000000
Successfully generated 2 random vectors of length: 100000000
Program execution time: 201834659 microseconds
```

Hình 3: ViDuHinhAnhTheoChieuNgang

2.5 Nhận xét

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

3 Thực hành nhóm

3.1 Yêu cầu

Thực hiện chương trình tính tích vô hướng của 2 vector.

3.2 Cơ sở toán học

Tích vô hướng của 2 vector là một phép toán đại số và cho kết quả là một số vô hướng, được tính bằng cách lấy tổng của tích từng cặp phần tử tương ứng:

$$\mathbf{u} \cdot \mathbf{v} = \sum_{i=1}^{n} u_i \cdot v_i$$

Trong đó:

- Với $\mathbf{u}$ và $\mathbf{v}$ là hai vector có cùng số chiều n.
- Với u_i và v_i là phần tử thứ i của $\mathbf{u}$ và $\mathbf{v}$ tương ứng.

Ví dụ:
Cho 2 vector $\mathbf{u} = \langle 1, 2, 3 \rangle$ và $\mathbf{v} = \langle 4, 5, 6 \rangle$, thì tích vô hướng của $\mathbf{u}$ và $\mathbf{v}$ là:

$$\mathbf{u} \cdot \mathbf{v} = 1 \cdot 4 + 2 \cdot 5 + 3 \cdot 6 = 42$$

3.3 Mã nguồn tính tích vô hướng của 2 vector

3.3.1 Mã nguồn tạo vector ngẫu nhiên

```
double sequentialDotProduct(double *vectorA, double *vectorB, int size)
{
    double result = 0.0;

    for (int i = 0; i < size; i++)
    {
        result += vectorA[i] * vectorB[i];
    }

    return result;
}
```

Mã nguồn 3: Mã nguồn tạo vector ngẫu nhiên

3.3.2 Mã nguồn lập trình tuần tự tính tích vô hướng của 2 vector

```
double sequentialDotProduct(double *vectorA, double *vectorB, int size)
{
    double result = 0.0;

    for (int i = 0; i < size; i++)
    {
        result += vectorA[i] * vectorB[i];
    }

    return result;
}
```

Mã nguồn 4: Mã nguồn lập trình tuần tự tính tích vô hướng của 2 vector

3.3.3 Mã nguồn lập trình song song tính tích vô hướng của 2 vector

```
double parallelDotProduct(double *vectorA, double *vectorB, int size)
{

    double result = 0.0;

    #pragma fomp parallel for reduction(+ : result)
    for (int i = 0; i < size; ++i)
    {
        result += vectorA[i] * vectorB[i];
    }

    return result;
}
```

Mã nguồn 5: Mã nguồn lập trình song song tính tích vô hướng của 2 vector

Trong mã nguồn sử dụng `#pragma omp parallel for reduction(+ : result)` là một chỉ thị được sử dụng trong OpenMP với ý nghĩa:

1. `#pragma omp`: thông báo cho trình biên dịch hiểu rằng phần mã nguồn sau đó sẽ được thực hiện theo kiểu lập trình đa luồng của OpenMP.
2. `parallel for`: thông báo cho trình biên dịch sử dụng đa luồng cho các lần lặp của vòng lặp `for` tiếp theo.
3. `reduction(+ : result)`: sử dụng để chỉ định một toán tử cụ thể. Khi một toán tử được chỉ định, mỗi luồng duy trì một bản sao riêng tư của biến và cuối vùng song song, các kết quả được kết hợp bằng cách sử dụng toán tử đã được chỉ định. Trong mã nguồn này, là phép cộng (+) trên biến `result`.

Trong ví dụ này, vòng lặp được lập trình đa luồng, và mỗi luồng đóng góp vào tổng của `result`. Cuối cùng, các kết quả cá nhân của luồng được kết hợp, và tổng cuối cùng được in ra màn hình.

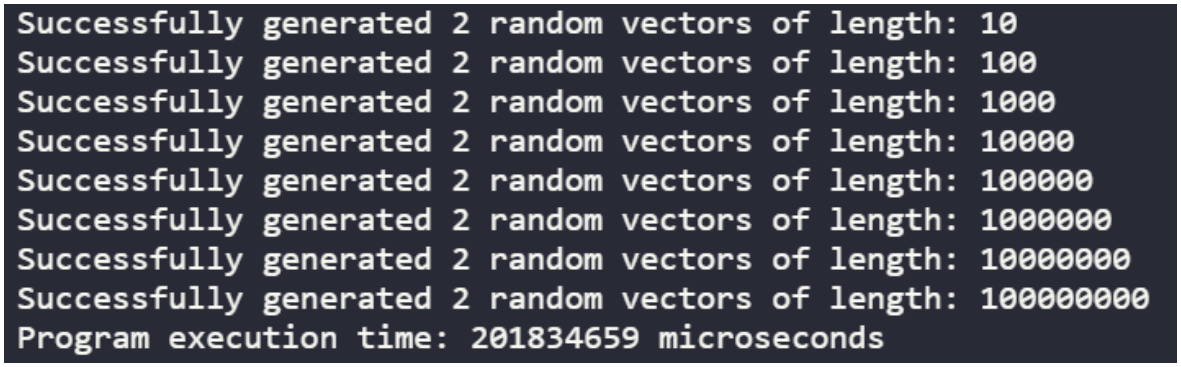
3.4 Kết quả:

3.4.1 Bảng kết quả tính tích vô hướng của 2 vector

BẢNG KẾT QUẢ														
Kích thước	Thời gian tuần tự	Song song 2 luồng		Song song 3 luồng		Song song 4 luồng		Song song 5 luồng		Song song 6 luồng		Song song 7 luồng	
		Thời gian	Hiệu suất	Thời gian	Hiệu suất	Thời gian	Hiệu suất	Thời gian	Hiệu suất	Thời gian	Hiệu suất	Thời gian	Hiệu suất
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113
11	112	113	113	113	113	113	113	113	113	113	113	113	113

Bảng 3: Bảng Kết Quả

3.4.2 Biểu đồ tính tích vô hướng của 2 vector



Hình 4: ViDuHinhAnhTheoChieuNgang

3.5 Nhận xét

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Kết luận

Báo cáo đã trình bày về tính toán song song sử dụng OpenMP cho phép tận dụng xử lý đồng thời với 2 bài toán:

- Thực hiện phép nhân ma trận với vector.
- Thực hiện chương trình tính tích vô hướng của 2 vector.

Việc sử dụng tính toán song song cũng đặt ra một số thách thức như đồng bộ hóa dữ liệu và quản lý tài nguyên.

Thông qua việc thực hiện báo cáo, các thành viên đã được rèn luyện kỹ năng làm việc nhóm và nâng cao khả năng hiểu biết về tính toán song song.

Trong quá trình hoạt động, đôi khi còn khó khăn. Kiến thức và khả năng trình bày của nhóm không tránh khỏi những thiếu sót, rất mong nhận được sự góp ý và bổ sung.

Tài liệu tham khảo

(1) https://en.wikipedia.org/wiki/Parallel_computing

(2) https://en.wikipedia.org/wiki/OpenMP

(3) https://en.wikipedia.org/wiki/Matrix_multiplication

(4) https://en.wikipedia.org/wiki/Dot_product

(5) https://hpc-tutorials.llnl.gov/openmp

(6) https://www.youtube.com/watch?v=JpU53_G-HAk

(7) https://hpc.llnl.gov/documentation/tutorials/introduction-parallel-computing-tutorial