

**ĐẠI HỌC QUỐC GIA TP HCM
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH**



Môn học: Tính Toán Song Song

BÀI TẬP LỚN

**NHÂN MA TRẬN TRÊN NHIỀU MÁY TÍNH
BẰNG CÁCH SỬ DỤNG ONE-SIDED COMMUNICATION**

GVHD: Nguyễn Quang Hùng

Danh sách thành viên:

Nguyễn Thị Xuân Mai	1811073
Tạ Minh Tâm	1811207
Hồ Anh Khiết	1710139

Mục Lục

1. Giới thiệu	3
2. MPI One-Sided Communication	4
3. Dữ liệu và hiện thực	5
3.1 Dữ liệu	5
3.2 Hiện thực	5
4. Kết quả và đánh giá	6
Tài liệu tham khảo	9

1. Giới thiệu

The Message Passing Interface Standard (MPI) là một tiêu chuẩn thông qua thư viện dựa trên sự đồng thuận của Diễn đàn MPI, có hơn 40 tổ chức tham gia, bao gồm các nhà cung cấp, nhà nghiên cứu, nhà phát triển thư viện phần mềm và người dùng. Mục tiêu của MPI là thiết lập một tiêu chuẩn di động, hiệu quả và linh hoạt để giao tiếp trong các mô hình song song. Những ưu điểm của MPI:

- Tiêu chuẩn hóa - MPI là thư viện thông qua duy nhất có thể được coi là một tiêu chuẩn. Nó được hỗ trợ trên hầu hết tất cả các nền tảng HPC.
- Tính di động - Có rất ít hoặc không cần phải sửa đổi mã nguồn của bạn khi bạn chuyển ứng dụng của mình sang một nền tảng khác hỗ trợ (và tuân thủ) tiêu chuẩn MPI.
- Hiệu suất - Việc triển khai MPI là của từng nhà cung cấp sẽ giúp khai thác các tính năng phần cứng gốc để tối ưu hóa hiệu suất. Việc hiện thực MPI là miễn phí.
- Chức năng - Có hơn 430 dịch vụ được xác định trong MPI-3, bao gồm phần lớn các thói quen trong MPI-2 và MPI-1.

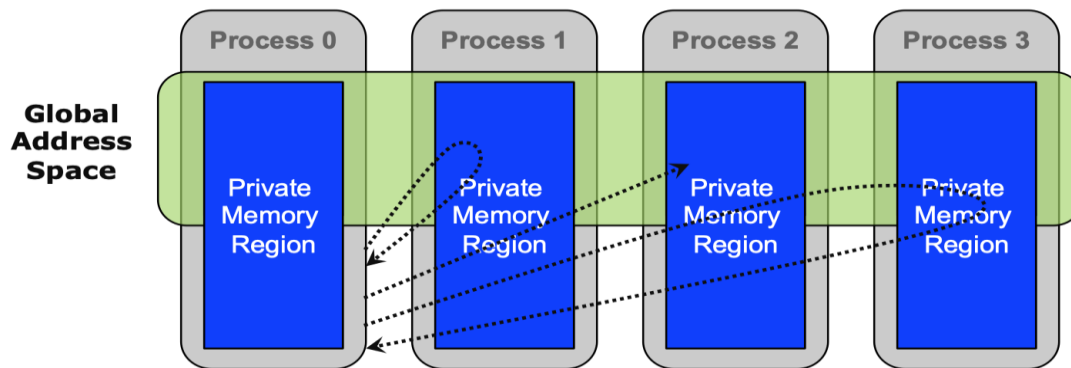
Chuẩn của MPI (Message Passing Interface) là Two-sided communication nghĩa là sẽ có một process gửi thông tin đi và một process nhận thông tin về. tuy nhiên điều này gây ra một hạn chế đó là process gửi thông tin sẽ bị delay cho tới khi process kia nhận được thông tin. Điều này sẽ gây ra tác hại lớn nếu process nhận bị delay quá lâu.

Từ đó MPI 2 RMA (Remote Memory Access) được ra đời hay còn được gọi là one-sided communication.

2. MPI One-Sided Communication

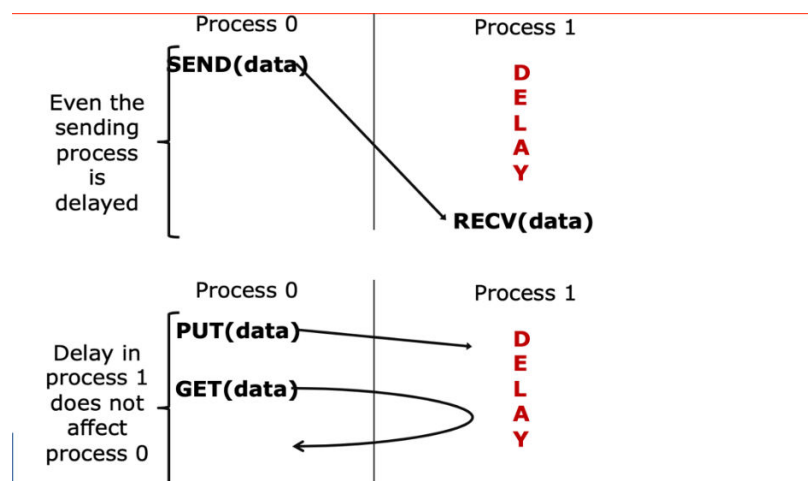
Ý tưởng cơ bản của One-sided communication model là tách rời việc truyền dữ liệu với việc đồng bộ quá trình:

- Có thể truyền đạt dữ liệu mà không cần yêu cầu về đồng bộ quá trình
- Mỗi process sẽ có một phần bộ nhớ để mở cho các processes khác truy cập.
- Các process khác có thể trực tiếp đọc hoặc ghi dữ liệu lên phần nhớ này.



Hình 1: Ý tưởng của One-sided communication¹.

Lợi ích của việc sử dụng One-sided communication so với Two-sided communication là hạn chế được khoảng thời gian bị delay của cả 2 process.



Hình 2: Lợi ích của One-sided communication²

¹ One-sided communication in MPI, William, <https://wgropp.cs.illinois.edu/courses/cs598-s16/lectures/lecture34.pdf>

² One-sided communication in MPI, William, <https://wgropp.cs.illinois.edu/courses/cs598-s16/lectures/lecture34.pdf>

Ý tưởng nhân ma trận sử dụng MPI One-sided communication:

- Tạo ra môi trường có nhiều process chạy song song với nhau.
- Mỗi process sẽ tạo ra một cửa sổ bộ nhớ chia sẻ (shared – memory).
- Ta sẽ lấy dữ liệu từ phần dùng chung đó ta thực hiện tính toán trong phần dữ liệu local.
- Sau khi tính toán xong ta ghi lại trên vùng dữ liệu shared để các process khác có thể sử dụng.

3. Dữ liệu và hiện thực

3.1 Dữ liệu

Dữ liệu là 3 cặp ma trận lần lượt có kích thước 100x100, 1.000x1.000, 10.000x10.000 được tạo bằng C. Có giá trị khởi tạo của mỗi phần tử là 1 để thuận tiện cho việc quan sát kết quả.

Các ma trận được lưu trong mục data tại link: [TTSS - Google Drive](#)

3.2 Hiện thực

Source code được lưu trong mục source tại link: [TTSS - Google Drive](#)

Hiện tại nhóm chưa được cấp tài khoản và không kết nối được với các máy khác. Vì vậy nhóm chỉ chạy local của node.

Video nhóm được lưu trong mục video tại link: [TTSS - Google Drive](#)

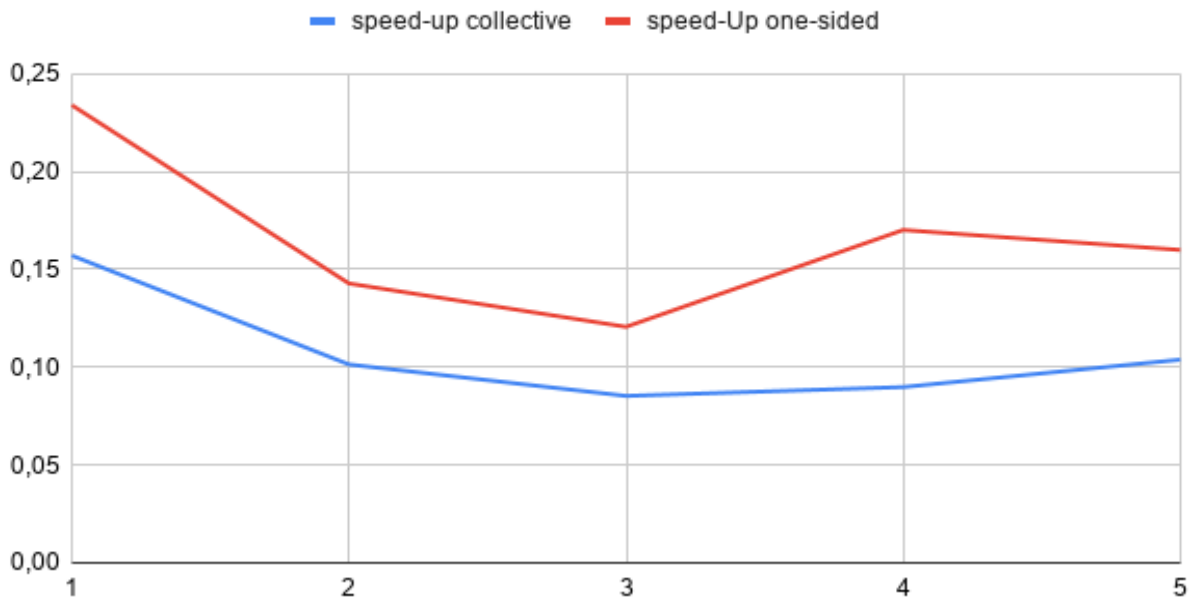
4. Kết quả và đánh giá

Hình chụp kết quả được lưu tại mục capture trong link: [TTSS - Google Drive](#)

Lần	Serial time (s)	Collective		One-sided (np = 5)	
		Time (s)	Speed-Up	Time (s)	Speed-Up
1	0,011	0,07	0,1571428571	0,047	0,2340425532
2	0,007	0,069	0,1014492754	0,049	0,1428571429
3	0,007	0,082	0,08536585366	0,058	0,1206896552
4	0,008	0,089	0,08988764045	0,047	0,170212766
5	0,008	0,077	0,1038961039	0,05	0,16
TB	0,0082	0,0774	0,1059431525	0,050	0,1633466135

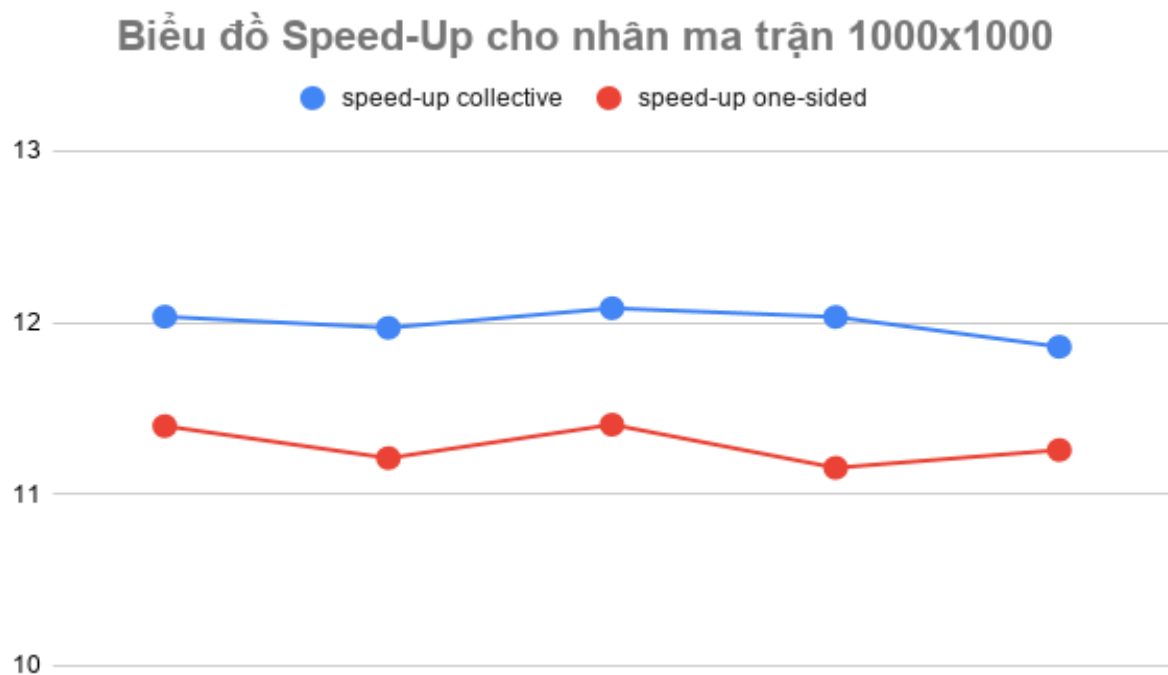
Bảng 1. Thời gian nhân hai ma trận 100x100

Biểu đồ Speed-Up cho nhân ma trận 100x100



Lần	Serial time (s)	Collective		One-sided (np = 20)	
		Time (s)	Speed-Up	Time (s)	Speed-Up
1	5,596	0,465	12,0344086	0,491	11,39714868
2	5,482	0,458	11,96943231	0,489	11,21063395
3	5,486	0,454	12,08370044	0,481	11,40540541
4	5,499	0,457	12,03282276	0,493	11,15415822
5	5,550	0,468	11,85897436	0,493	11,25760649
TB	5,5226	0,4604	11,99522155	0,4894	11,28442991

Bảng 2. Thời gian nhân hai ma trận 1000x1000



Lần	Serial time (s)	Collective		One-sided (np = 25)	
		Time (s)	Speed-Up	Time (s)	Speed-Up
1	∞	0,465	∞	711,807	∞
2	∞	0,458	∞	715,259	∞
3	∞	0,454	∞	711,211	∞
4	∞	0,457	∞	717,124	∞
5	∞	0,468	∞	712,666	∞
TB	∞	0,4604	∞	713,6134	∞

Bảng 3. Thời gian nhân hai ma trận 10000x10000

Đánh giá: Do kích thước ma trận 10000x10000 quá lớn nên việc nhân ma trận tuần tự sẽ tốn thời gian khá lớn. Vì vậy ta chỉ có thể ước lượng thời gian là ∞ . Do đó Speed-Up sẽ rất lớn.

Ma trận	100x100	1000x1000	10000x10000
One-sided (np=25) (s)	0,0738	0,5918	713,6134
One-sided (min) (s)	Np = 5 0,050	Np = 20 0,4894	Np = 25 713,6134

Bảng 4. Thời gian trung bình nhân ma trận với np=25 và trường hợp np tốt nhất.

Đánh giá kết quả:

- Trường hợp nhân hai ma trận 100×100 , việc thực hiện tuần tự lại tốn ít thời gian hơn khi sử dụng giải thuật song song (collective và one-sided) do với số lượng công việc cần tính toán nhỏ, việc giao tiếp giữa các process có thể chiếm thời gian lớn hơn việc tính toán.
 - Trường hợp nhân hai ma trận 1000×1000 , chúng ta đã thấy rõ hiệu quả của việc thực hiện giải thuật song song do số lượng công việc cần tính toán rất lớn.
 - Trường hợp nhân hai ma trận 10000×10000 , việc tính toán tuần tự tốn quá nhiều thời gian, chỉ có thể đo được thời gian tính toán khi sử dụng giải thuật song song
- ⇒ Nhìn chung, đối với việc thực hiện nhiều phép tính toán chúng ta cần sử dụng các giải thuật song song để rút ngắn thời gian. Tùy vào số lượng công việc cũng như giải thuật để xác định số process thích hợp để tối ưu và đạt được hiệu quả. Không nhất định sử dụng nhiều process thì hiệu quả cao hơn.

Tài liệu tham khảo

- Index of /static/docs/v3.3/www3
Nguồn: <https://www.mpich.org/static/docs/v3.3/www3/>
Lần truy cập cuối: 17/12/2020
- MPI topic: One-sided communication
Nguồn: https://pages.tacc.utexas.edu/~eijkhout/pcse/html/mpi-onesided.html?fbclid=IwAR0LKvU6I7qVL5a2unR9mWqzQWgR4ul7eJZH5uWhXDnc_1ozkPmaFwecUS0
Lần truy cập cuối: 17/12/2020
- One-sided communication in MPI, William
Nguồn: <https://wgropp.cs.illinois.edu/courses/cs598-s16/lectures/lecture34.pdf>
Lần truy cập cuối: 17/12/2020