

Lab 2 Parallel Programming with MPI

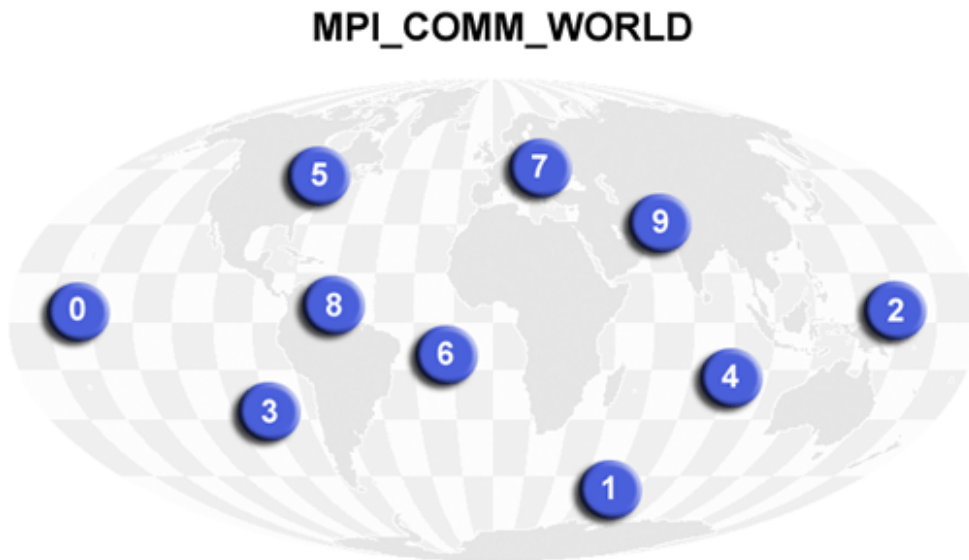
Point-to-Point Communication (1)

1. Mục tiêu

- SV tìm hiểu và sử dụng các hàm truyền thông gửi/nhận dữ liệu trong thư viện MPI
- Một số hàm giao tiếp nhóm SV cần tìm hiểu :
 - MPI_Init(), MPI_Comm_rank, MPI_Comm_size, MPI_Finalize
 - MPI_Send, MPI_Ssend, MPI_Bsend, MPI_Rsend, MPI_Isend
 - MPI_Recv, MPI_Irecv
 - MPI_Wtime(), MPI_Get_count, MPI_Get_elements, MPI_Wait, MPI_Test

2. Nội dung

2.1 Giới thiệu



- Sự giao tiếp giữa 1 nhóm process trong cùng communicator
- Mỗi process đều phải gọi hàm giao tiếp nhóm
- SV tìm hiểu xem mỗi hàm giao tiếp nhóm có chức năng gì và thực hiện các chương trình mẫu trong mục 2.2

2.2 Giới thiệu các hàm

Danh sách các hàm trong thư viện lập trình MPI tra cứu tại đây:

<http://www.mcs.anl.gov/research/projects/mpi/www/www3/>

MPI_Send()

- Performs a blocking send

MPI_Get_count()

- Gets the number of "top level" elements

2.2 Một số chương trình minh họa

2.2.1 Chương trình Hello world

```
#include <mpi.h>
#include <stdio.h>

int main(int argc, char* argv[]){
    int rank, size;
    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Comm_size(MPI_COMM_WORLD, &size);
    printf("Hello MPI from process rank: %d of %d \n", rank, size);
    MPI_Finalize();
    return 0;
}
```



THỰC HÀNH:

Gõ chương trình trên vào file 'hellompi.c'. Biên dịch và chạy chương trình helloworld:

```
$ vi hellompi.c
```

```
$ mpicc hellompi.c -o hellompi
```

```
$ mpirun -np 2 hellompi
```

Kết quả in ra:

```
Hello MPI from process rank: 0 of 2  
Hello MPI from process rank: 1 of 2
```

2.2.2 Chương trình sử dụng pPoint.c

```
#include <mpi.h>  
#include <stdio.h>  
#include <stdlib.h>  
  
int main(int argc, char **argv){  
    int rank,size, tag;  
    double a,b,s;  
    double t;  
    MPI_Status status;  
    MPI_Init(&argc,&argv);  
    MPI_Comm_rank(MPI_COMM_WORLD,&rank);  
    MPI_Comm_size(MPI_COMM_WORLD,&size);  
    fprintf(stdout,"\n Process %d of %d processes starts \n", rank, size);  
    t = MPI_Wtime();  
    if(rank == 1) {  
        b = 12.2;  
        MPI_Recv(&a,1,MPI_DOUBLE,MPI_ANY_SOURCE,MPI_ANY_TAG,MPI_COMM_WORLD, &status);  
        s = b - a ;  
        fprintf(stdout," The result is : %f \n", s);  
    }  
    else {  
        a = rank; tag=1;  
        MPI_Send(&a, 1, MPI_DOUBLE, 1, tag, MPI_COMM_WORLD);
```

```
}  
  
t = MPI_Wtime() - t;  
  
printf("Round time: %lf (ns) \n", t);  
  
MPI_Finalize();  
  
return 0;  
  
}
```



THỰC HÀNH:

Câu hỏi:

☺ Sinh viên hãy thực thi bài này nhiều lần với số process khác nhau (vd: 2, 3, 4...) và nhận xét về kết quả in ra có giống nhau ở nhiều lần chạy không? Ví dụ:

```
$ mpirun -np 2 pPoint
```

```
$ mpirun -np 3 pPoint
```

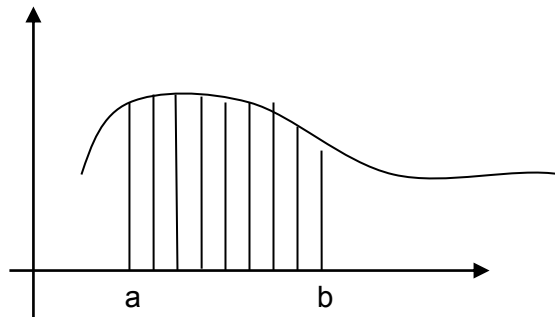
```
$ mpirun -np 4 pPoint
```

3. Bài tập

SV hiện thực các bài tập:

Bài 3.1. Viết chương trình nhân hai vector.

Bài 3.2. Tính tích phân của hàm $f(x) > 0$ và liên tục trong khoảng $[a, b]$ bằng phương pháp chia miền này thành N hình thang nhỏ. Sai số tùy thuộc vào số lượng hình thang này.



LƯU Ý: SV PHẢI NỘP SOURCE CODE CÁC BÀI TẬP LÊN SAKAI ĐÚNG HẠN