Các thao tác trên communicator

Hà nội, 6/2008

Đại học Bách khoa Hà Nội

Center of High Performance Computing

Hanoi University of Technology

{hpcc@mail.hut.edu.vn}

Nội dung bài học

- Communicator
- Các loại communicator
- Thao tác với intra-communicator
- Thao tác với inter-communicator

Communicator

- Một communicator (comm) bao gồm một tập các tiến trình, hay một nhóm các tiến trình.
- Truyền thông giữa các tiến trình thực hiện trên ngữ cảnh communicator
- MPI_COMM_WORLD:
  - Là communicator mặc định khi khởi tạo chương trình
  - Cho phép các tiến trình truyền thông điểm điểm hoặc cộng tác
- Một số ứng dụng đòi hỏi truyền thông giữa một nhóm nhỏ các tiến trình.

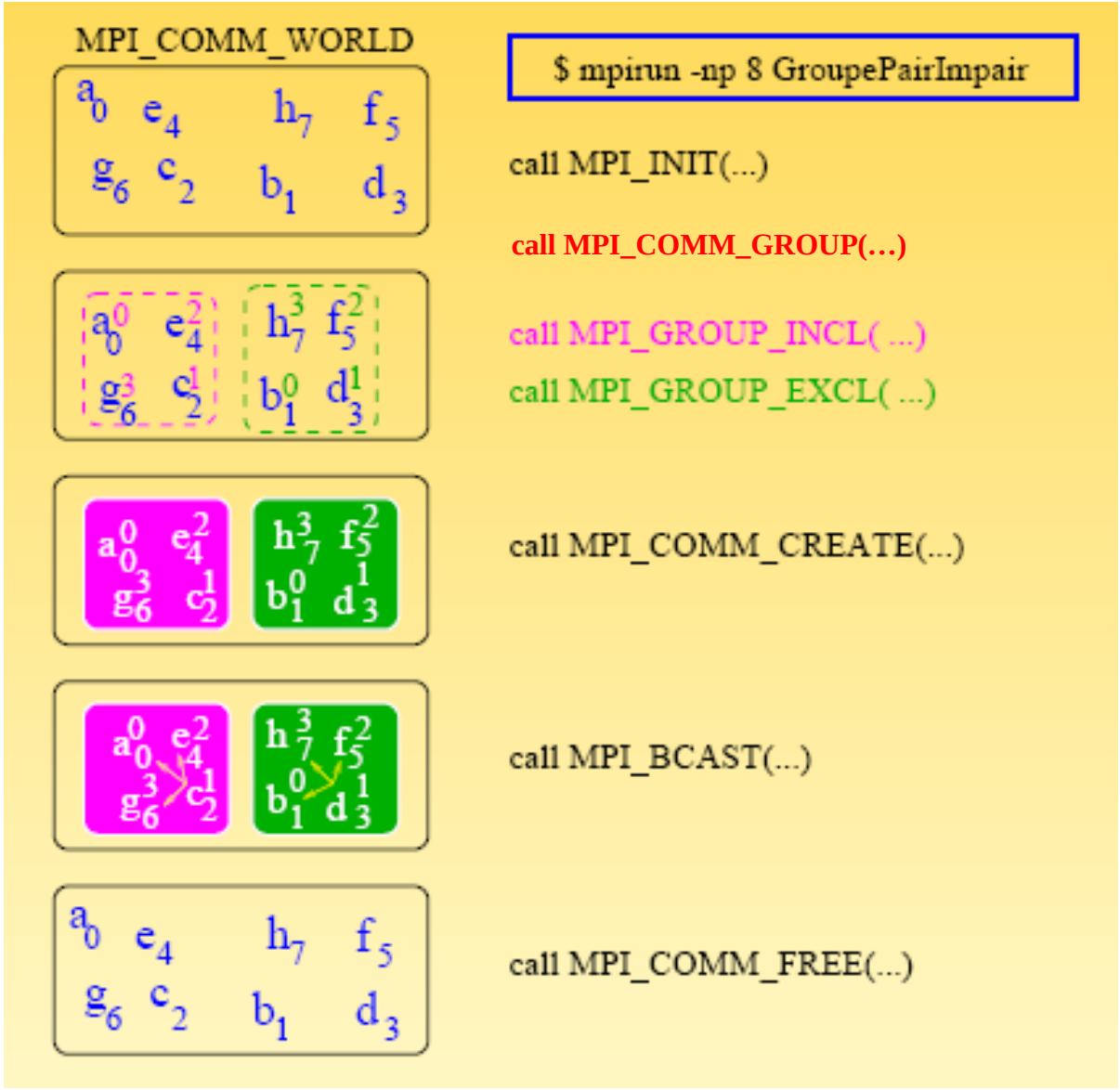
Communicator

- Một communicator gồm:
  - Một nhóm các tiến trình
  - Một ngữ cảnh (context) hạ tầng trao đổi thông tin, tạo ra khi comm được thiết lập
- Nhóm = Tập hợp các tiến trình
- Mỗi communicator có một định danh nhóm hay một group handle liên kết với nó.
- Group handle sẽ được dùng cho các thao tác thêm bớt tiến trình, và tạo ra communicator

Các loại communicator

- Intra-communicators:
  - Liên quan đến các truyền thông giữa các tiến trình trong cùng một communicator
  - Về cơ bản, intra-communicator gồm tập con các tiến trình của MPI_COM_WORLD
  - Thường xử lý các tác vụ: xử lý dòng, cột, ma trận con của ma trận, tăng tính trong sáng của chương trình,...
- Inter-communicators:
  - Liên quan đến truyền thông giữa các intra-communicator
- Có 2 cách tạo ra communicator:
  - Tạo communicator từ group (nhóm tiến trình)
  - Tạo communicator từ communicator.

Qui trình tạo communicator từ nhóm



Định tuyến MPI_Comm_group

```
int MPI_Comm_group( MPI_Comm comm, MPI_Group *group )
```

Variable Name	C Type	In/Out	Description
comm	MPI_Comm	Input	Communicator handle
group	MPI_Group *	Output	Group handle

- Lấy về group handle của communicator

Ví dụ lấy về group handle

```
#include "mpi.h"
MPI_Comm  comm_world;
MPI_Group group_world;

comm_world = MPI_COMM_WORLD;
MPI_Comm_group(comm_world, &group_world);
```

Định tuyến MPI_group_incl

- Tạo ra nhóm mới từ nhóm đã tồn tại bằng cách chỉ định các tiến trình sẽ là thành viên nhóm mới

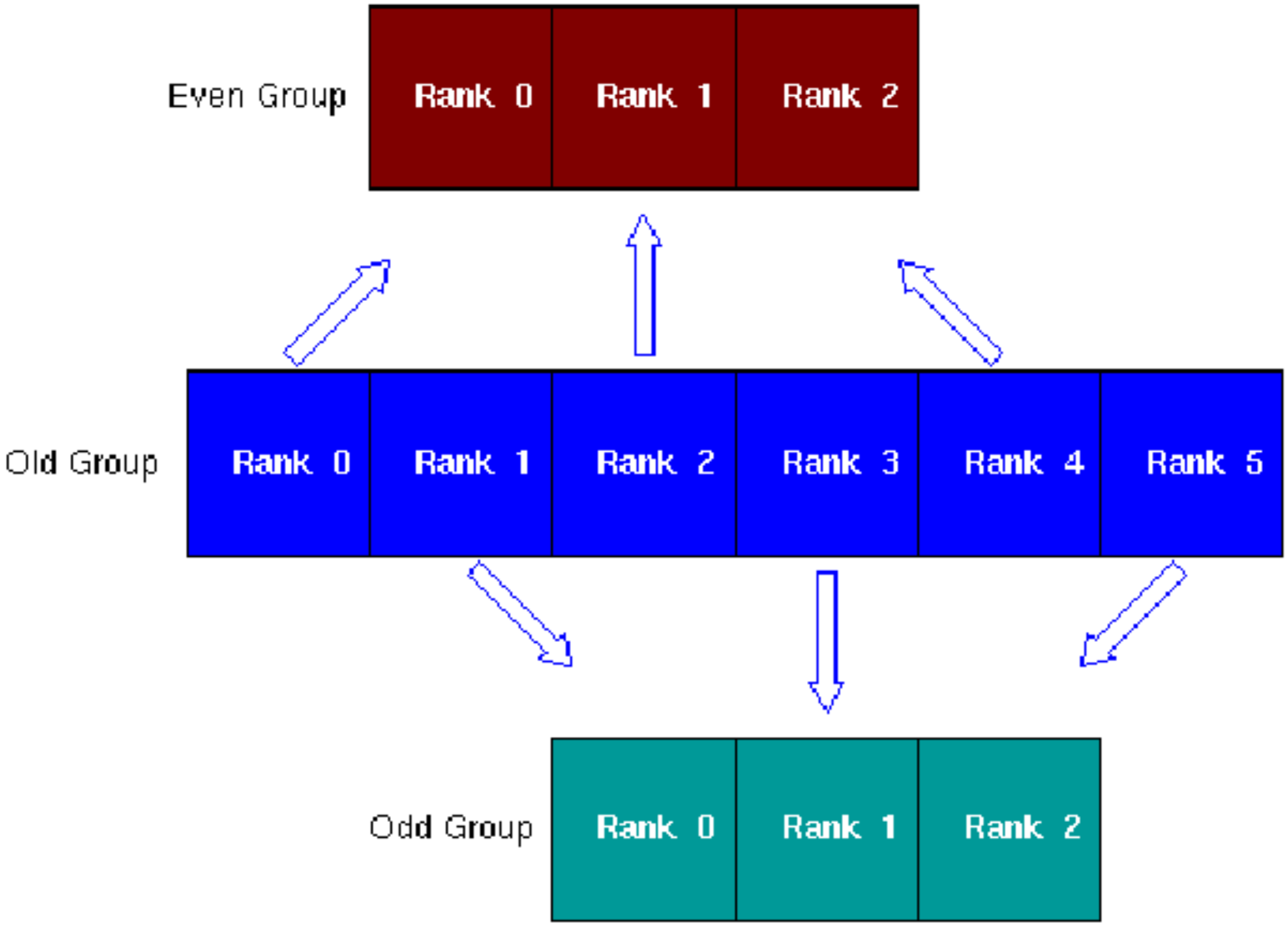
```
int MPI_Group_incl( MPI_Group old_group, int count, int *members,
MPI_Group *new_group )
```

Variable Name	C Type	In/Out	Description
old_group	MPI_Group	Input	Group handle
count	int	Input	Number of processes in new_group
members	int *	Input	Array of size count defining process ranks (in old_group) to be included (in new_group)
new_group	MPI_Group *	Output	Group handle

Ví dụ tạo nhóm tiến trình

```
#include "mpi.h"
MPI_Group group_world, odd_group, even_group;
int i, p, Neven, Nodd, members[8], ierr;

MPI_Comm_size(MPI_COMM_WORLD, &p);
MPI_Comm_group(MPI_COMM_WORLD, &group_world);

Neven = (p+1)/2;     /* processes of MPI_COMM_WORLD are divided */
Nodd = p - Neven;    /* into odd- and even-numbered groups */
for (i=0; i<Neven; i++) {    /* "members" determines members of
even_group */
  members[i] = 2*i;
};

MPI_Group_incl(group_world, Neven, members, &even_group);
```

Ví dụ tạo nhóm tiến trình



Định tuyến MPI_Group_incl

- Vị trí của tiến trình gọi trong nhóm mới được xác định bởi mảng member
  - Tiến trình gọi có rank là member(i) sẽ có rank là i trong nhóm mới
  - i nằm trong khoảng (0, count -1)
- Nếu count = 0, định danh của nhóm mới có giá trị MPI_GROUP_EMPTY
- Hai nhóm có thể có tiến trình giống nhau, nhưng khác nhau về thứ tự phụ thuộc vào ma trận member

Định tuyến MPI_group_excl

- Tạo ra nhóm mới từ nhóm đã tồn tại bằng cách chỉ định các tiến trình không phải thành viên nhóm mới

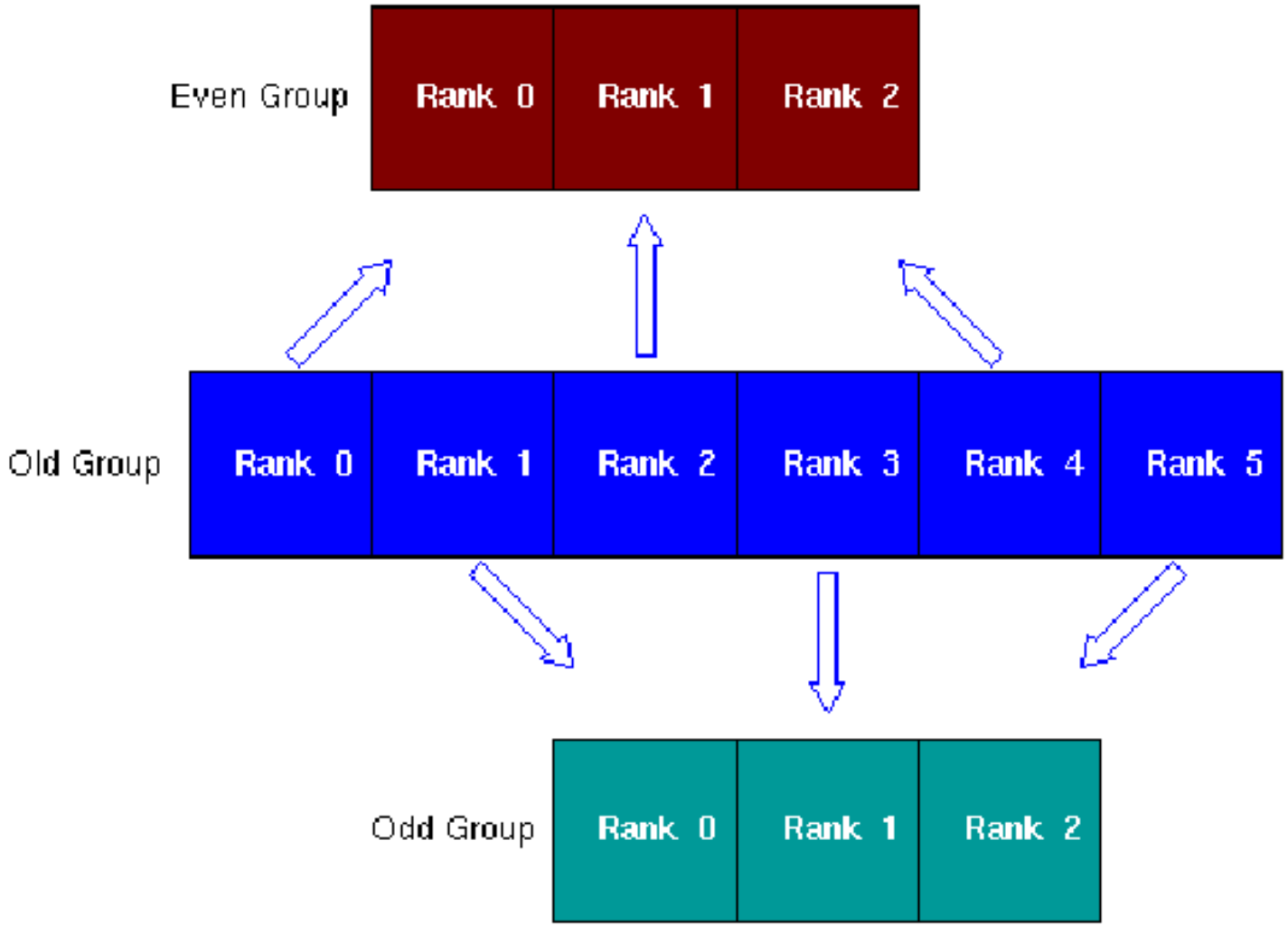
```
int MPI_Group_excl( MPI_Group group, int count, int *nonmembers,
MPI_Group *new_group )
```

Variable Name	C Type	In/Out	Description
group	MPI_Group	Input	Group handle
count	int	Input	Number of processes in nonmembers
nonmembers	int *	Output	Array of size count defining process ranks to be excluded
new_group	MPI_Group *	Output	Group handle
ierr	See (*)	Output	Error flag

Ví dụ tạo nhóm mới

```
#include "mpi.h"
MPI_Group group_world, odd_group, even_group;
int i, p, Neven, Nodd, nonmembers[8], ierr;

MPI_Comm_size(MPI_COMM_WORLD, &p);
MPI_Comm_group(MPI_COMM_WORLD, &group_world);

Neven = (p+1)/2;     /* processes of MPI_COMM_WORLD are divided */
Nodd = p - Neven;    /* into odd- and even-numbered groups */
for (i=0; i<Neven; i++) {    /* "nonmembers" are even-numbered procs
*/
  nonmembers[i] = 2*i;
};

MPI_Group_excl(group_world, Neven, nonmembers, &odd_group);
```

Ví dụ tạo nhóm mới



Định tuyến MPI_group_excl

- Vị trí tiến trình gọi trong nhóm mới tương ứng với rank của nó trong nhóm cũ
- Thứ tự các phần tử trong mảng nonmember không ảnh hưởng đến rank của tiến trình trong nhóm mới
- Nếu count=0, nhóm mới tạo ra giống hệt nhóm cũ
- Các rank của các tiến trình trong mảng nonmember:
  - Không được trùng nhau.
  - Phải hợp lệ (tồn tại)

Các định tuyến lấy thông tin nhóm

- Lấy về rank của tiến trình trong nhóm
  - MPI_Group_rank (MPI_Group group, int *rank);
- Lấy về kích thước của nhóm tiến trình
  - MPI_Group_size (MPI_Group group, int *size);
- So sánh mối quan hệ giữa hai nhóm tiến trình
  - MPI_Group_compare (MPI_Group group1, MPI_Group group2, int *result);
  - Giá trị trả về, biến result:
    - MPI_IDENL: các tiến trình trong hai nhóm giống nhau, được đánh thứ tự rank giống nhau
    - MPI_SIMILAR: các tiến trình trong hai nhóm giống nhau, được đánh thứ tự rank khác nhau
    - MPI_UNEQUAL: mỗi quan hệ khác

Định tuyến MPI_Group_rank

```
#include "mpi.h"
MPI_Group group_world, worker_group;
int i, p, ierr, group_rank;

MPI_Comm_size(MPI_COMM_WORLD, &p);
MPI_Comm_group(MPI_COMM_WORLD, &group_world);
MPI_Group_excl(group_world, 1, 0, &worker_group);

MPI_Group_rank(worker_group, &group_rank);
```

- Trước khi tạo nhóm
  - MPI_COM_WORLD: (0, 1, …, p-1)
  - Worker_group:
- Sau khi tạo nhóm
  - MPI_COM_WORLD: 0
  - Worker_group: (0, 1, …, p-2)
- Nếu tiến trình đang gọi có rank = 0: giá trị group_rank là MPI_UNDEFINED

Định tuyến MPI_Group_free

```
int MPI_Group_free( MPI_Group *group )
```

- Trả group về cho hệ thống
- Không giải phóng communicator chứa đựng group đó.
- Dùng định tuyến MPI_Comm_free để giải phóng communicator

```
#include "mpi.h"
MPI_Group group_world, worker_group;
int i, p, ierr, group_rank;

MPI_Comm_size(MPI_COMM_WORLD, &p);
MPI_Comm_group(MPI_COMM_WORLD, &group_world);
MPI_Group_excl(group_world, 1, 0, &worker_group);
MPI_Group_rank(worker_group, &group_rank);

MPI_Group_free(worker_group);
```

MPI_Comm_Create

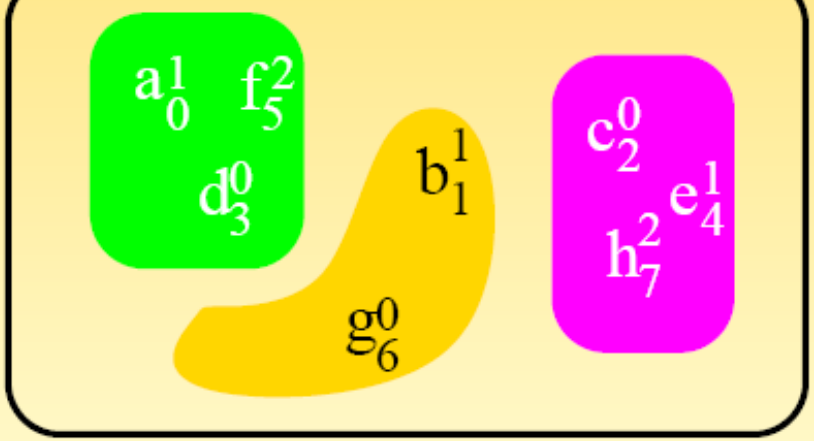
- Tạo ra communicator từ một communicator đã có sẵn
- Tất cả các tiến trình cần gọi một hàm giống nhau với các tham số giống nhau
- Các tiến trình không tham dự trong nhóm cho kết quả MPI_Comm_Null
- Giải phóng bằng MPI_Comm_Free

```
#include "mpi.h"
MPI_Comm  comm_world, comm_worker;
MPI_Group group_world, group_worker;
int ierr;

comm_world = MPI_COMM_WORLD;
MPI_Comm_group(comm_world, &group_world);
MPI_Group_excl(group_world, 1, 0, &group_worker);  /* process 0 not
member */

MPI_Comm_create(comm_world, group_worker, &comm_worker);
```

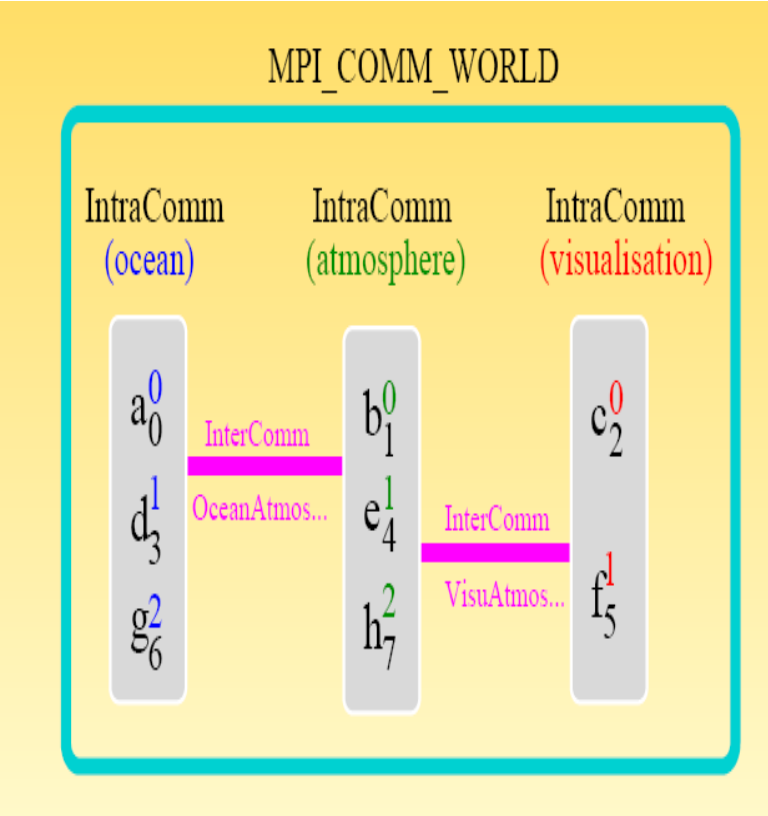
Tạo một comm từ một comm khác: MPI_Comm_Split

```
int MPI_Comm_split( MPI_Comm old_comm, int color, int key, MPI_Comm
*new_comm )
```

rang	0	1	2	3	4	5	6	7
processus	a	b	c	d	e	f	g	h
couleur	0	2	3	0	3	0	2	3
clef	2	15	0	0	1	3	11	1



Truyền thông giữa các inter-communicator



- Mục đích inter-communicator
  - Kết nối hạ tầng truyền thống giữa hai nhóm rời nhau
- Cấu trúc của communicator:
  - Tách thành nhóm cục bộ và nhóm từ xa
  - Có tính đối xứng

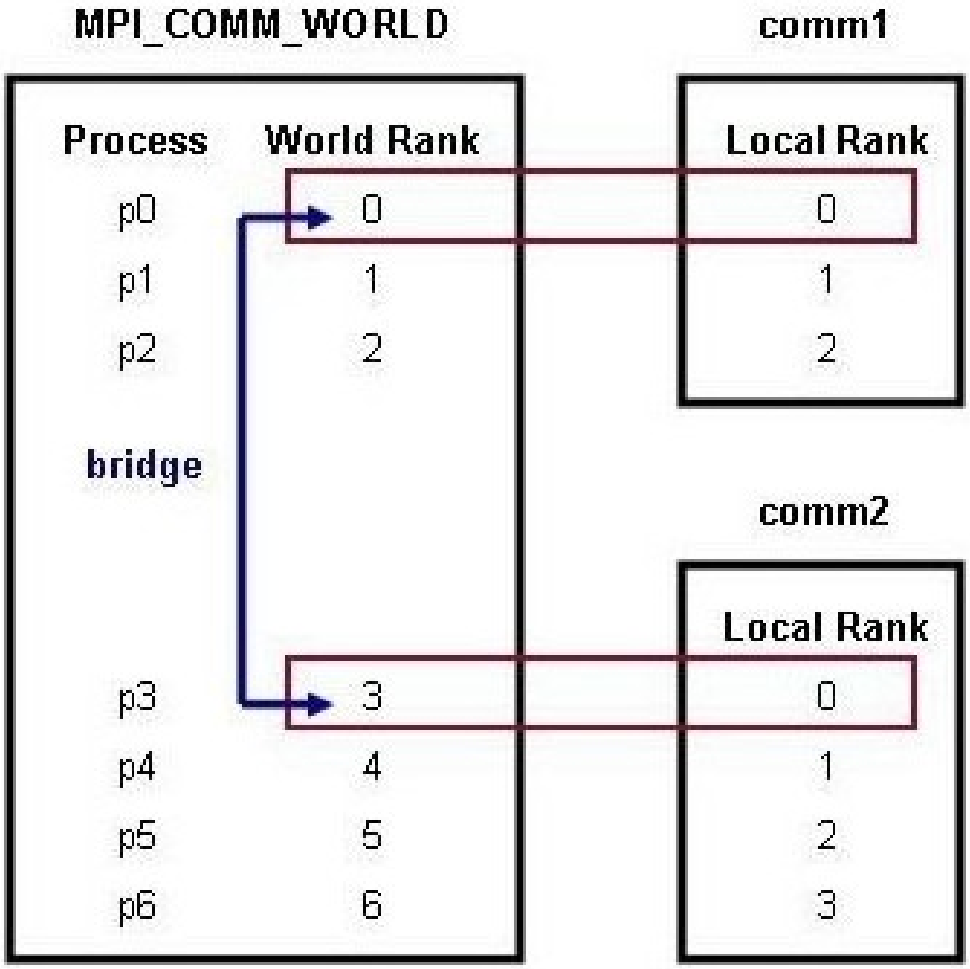
- Kiểm tra communicator là intra hay inter, dùng định tuyến:
  - int MPI_Comm_test_inter (MPI_Comm comm, int *flag)

Tạo inter-communicator từ hai intra-communicator

```
MPI_Intercomm_create (MPI_Comm local_comm, int
   local_leader, MPI_Comm peer_comm, int remote_leader,
   int tag, MPI_Comm *intercomm_out)
```

- Tạo inter-communicator bằng cách kết nối hai intra-communicator:
  - Tiến trình có rank local_leader trong nhóm local_comm truyền thông với một tiến trình remote_leader trong nhóm peer_comm
  - Hai tiến trình trong hai nhóm tạo thành một cầu nối hai intra-communicators
  - Inter-communicator được tạo ra xác định bởi con trỏ intercomm_out
- Đây là định tuyến truyền thông cộng tác:
  - Phải được gọi bởi các tiến trình tham gia vào định tuyến

Ví dụ tạo inter-communicator



Tạo inter-communicator từ inter-communicator có sẵn

MPI_Intercomm_merge (MPI_Comm intercomm, int high, MPI_Comm *newintercomm);

- Tổ hợp nhóm gồm 2 nhóm trong intercomm kết hợp để tạo thành newintercomm
- Giá trị *high* xác định thứ tự bên trong tổ hợp nhóm
  - Tất cả các tiến trình trong một nhóm phải truyền giá trị high giống nhau
  - Nhóm có giá trị *high* = *false* sẽ được xếp trước nhóm có giá trị *high* = *true* trong tổ hợp nhóm
  - Nếu tất cả các tiến trình có giá trị high giống nhau thì thứ tự sẽ được phân phối tùy ý

Kết luận

- Nhóm và ngữ cảnh truyền thông tạo ra các communicator
- Communicator định nghĩa khả năng trao đổi thông tin giữa các tiến trình
- Communicator được sử dụng để tách bạch hạ tầng truyền thông
- Tất cả các quá trình truyền tin đều cần đến 1 communicator
- Cho phép tránh nhầm lẫn khi lựa chọn các thông báo.
- Cho phép thực hiện việc lập trình cấu trúc trong hệ thống song song, tách biệt các modul.