Giới thiệu về lập trình song song theo mô hình truyền thông điệp

Hà nội, 6/2008

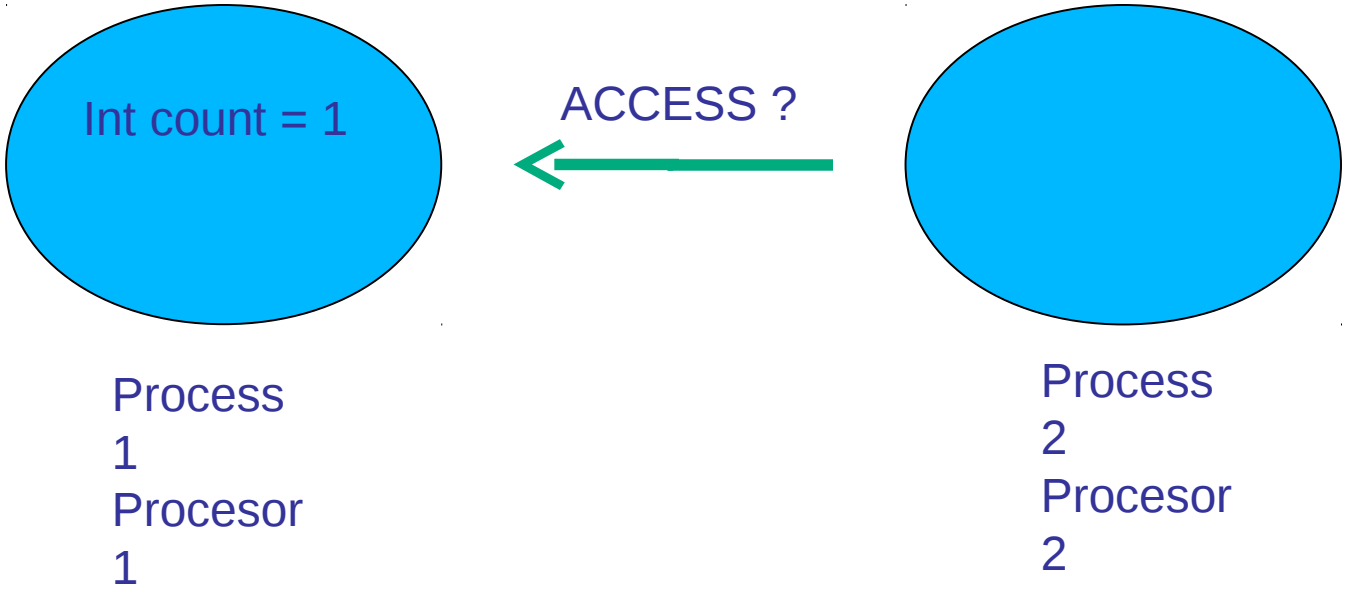
Đại học Bách khoa Hà Nội

Center of High Performance Computing
Hanoi University of Technology
{hpcc@mail.hut.edu.vn}

Nội dung bài học

- Mô hình truyền thông điệp
- Cấu trúc chương trình truyền thông điệp
- Các khái niệm trong chương trình
- Chuẩn bị môi trường lập trình
- Biên dịch, chạy chương trình song song
- Demo

Mô hình truyền thông điệp (Message Passing Model)



Cần cơ chế để các tiến trình truy cập bộ nhớ của nhau

Mô hình truyền thông điệp: Chia sẻ dữ liệu giữa các tiến trình bằng cách truyền thông điệp.

Chuẩn MPI (Message Passing Interface)

http://www.mpi-forum.org

- Chuẩn MPI là kết quả sau hơn 2 năm thảo luận của MPI Forum, 1 nhóm gồm khoảng 60 người từ 40 tổ chức khác nhau đại diện cho những nhà phân phối các hệ thống song song, những phòng thí nghiệm quốc gia và những trường đại học danh tiếng

- MPI là một thư viện các hàm có thể chèn vào mã nguồn để truyền dữ liệu giữa các tiến trình

Chuẩn MPI

- Chuẩn MPI-1:
  - Xác định quy cách đặt tên, quá trình thực hiện và kết quả trả về của các hàm/chương trình con trong thư viện. Mọi hàm/chương trình con trong thư viện đều phải tuân theo những quy tắc trên, điều này đảm bảo khả năng khả chuyển của các chương trình viết theo chuẩn MPI khi biên dịch và chạy trên các kiến trúc máy tính, hệ điều hành khác nhau.
  - Chi tiết cài đặt của các hàm/chương trình con trong thư viện là do những nhà cung cấp quyết định (vd: trường Đại học Indiana, Hoa Kỳ phát triển LAM/MPI, MPICH, OPENMPI,...)
  - Quá trình cài đặt các hàm/chương trình con trong thư viện phải tương thích với hầu hết các kiến trúc máy tính, hệ điều hành phổ biến.

- Năm 1994: MPI-1.0; hiện nay mới nhất là MPI-2.1
  - Chuẩn MPI-2.1 định nghĩa thêm các định tuyến nâng cao: vào ra song song, truyền thông 1 phía, ...
  - Tuy nhiên chưa được hỗ trợ nhiều

Khi nào dùng và không dùng MPI

- MPI dùng khi cần viết chương trình:
  - Đoạn mã song song có tính khả chuyển thông qua các flatform khác nhau
  - Cần đạt hiệu năng cao
  - Cần viết thư viện song song
- Khi nào không nên dùng MPI:
  - Có thể đạt hiệu năng bằng cách dùng các hàm song song sẵn có của thư viện Fortran hoặc của OpenMP
  - Đã có các thư viện song song viết bằng MPI, như thư viện toán học song song
  - Không cần đoạn mã nào chạy song song

Cấu trúc chương trình song song theo chuẩn MPI

- Ngôn ngữ C

Định nghĩa các prototype: Hàm, macro, hằng,….

Khởi tạo môi trường MPI

- Dọn dẹp dữ liệu MPI
- Hủy bỏ các hàm đang chạy

```
#include<mpi.h>


main(int argc,char* argv[])
{
   intitation of variable;
   MPI_Init(&argc,&argv);
   ....
   ...
   ...
   MPI_Finalize();
}
```

Các đặc điểm cơ bản của CT truyền thông điệp

- Bao gồm nhiều instance của 1 chương trình tuần tự.
- Các instance truyền thông với nhau qua các định tuyến trong thư viện MPI

- Các định tuyến được chia thành:
  - Khởi tạo, quản lý và ngắt qúa trình truyền thông
  - Truyền thông giữa các cặp tiến trình(truyền thông điểm – điểm)
  - Truyền thống giữa các nhóm nhóm(truyền thông công cộng)
  - Tạo các loại dữ liệu có cấu trúc riêng
  - Thao tác trên communicator

Quy ước đặt tên

- Tất cả phần tử trong MPI (định tuyến, biến, hằng, ...) đều bắt đầu bằng từ MPI_
- Tên định tuyến:
  - MPI_Xxxxxx(tham số,...)
    - MPI_Init($argc, $args)
- Tên hằng:
  - MPI_XXXXX
  - MPI_COMM_WORLD, MPI_REAL

Định tuyến MPI và giá trị trả về

- Các định tuyến MPI được thực thi như hàm trong C
- Giá trị trả về của định tuyến là 1 số nguyên (integer) cho biết trạng thái kết thúc của hàm (lỗi hay thành công)

```
int err;
err = MPI_Init($argc, $args);
if (err == MPI_SUCCESS)
{

}
```

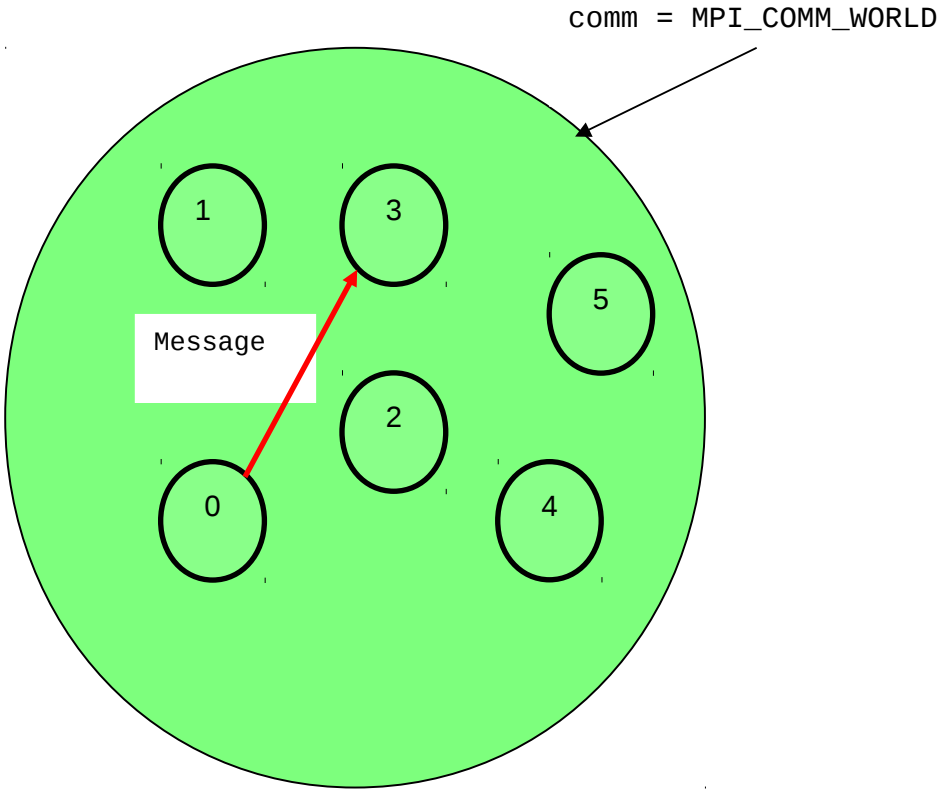
Các kiểu dữ liệu của chuẩn MPI

- Kiểu dữ liệu khai báo là MPI_Datatype
- Ngoài ra còn có các kiểu dữ liệu dẫn xuất từ kiểu dữ liệu chuẩn

Kiểu dữ liệu MPI	Kiểu dữ liệu tương ứng trong C
MPI_CHAR	signed char
MPI_SHORT	signed short int
MPI_INT	signed int
MPI_LONG	signed long int
MPI_UNSIGNED_CHAR	unsigned char
MPI_UNSIGNED_SHORT	unsigned short int
MPI_UNSIGNED	u
MPI_UNSIGNED_LONG	unsigned long int
MPI_FLOAT	float
MPI_DOUBLE	double
MPI_LONG_DOUBLE	long double
MPI_PACKED	(none)
MPI_BYTE	(none)

Communicator

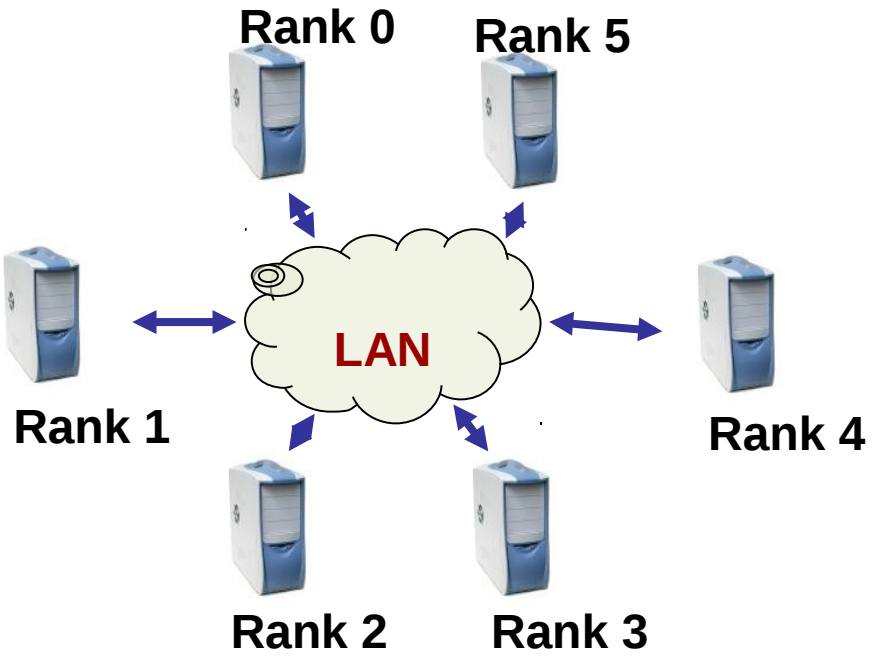
- Communicator: Một nhóm các tiến trình có thể truyền thống với nhau
- Một tiến trình có thể thuộc nhiều Communicator



- Kiểu dữ liệu biểu diễn Communicator: MPI_Comm
- MPI_Comm comm: xác định communicator của node gửi và nhận
- MPI_COM_WORLD: Mọi tiến trình đều thuộc communicator này

Rank

- Mỗi tiến trình trong 1 communicator có 1 định danh, gọi là Rank, đánh số bắt đầu từ 0.
- Một tiến trình có thể các rank khác nhau khi thuộc về các communicator khác nhau



Các định tuyến thiết lập môi trường và lấy các thông số về môi trường

- int MPI_Init()
  - Khởi tạo tham số cho môi trường MPI
- int MPI_Comm_rank(MPI_Comm comm, int *rank)
  - Trả về rank của tiến trình
- int MPI_Comm_size(MPI_Comm comm, int *size);
  - Trả về số tiến trình trong comm
- int MPI_Finalize()
  - Giải phóng dữ liệu, ngắt các định tuyến MPI

Cài đặt môi trường thử nghiệm

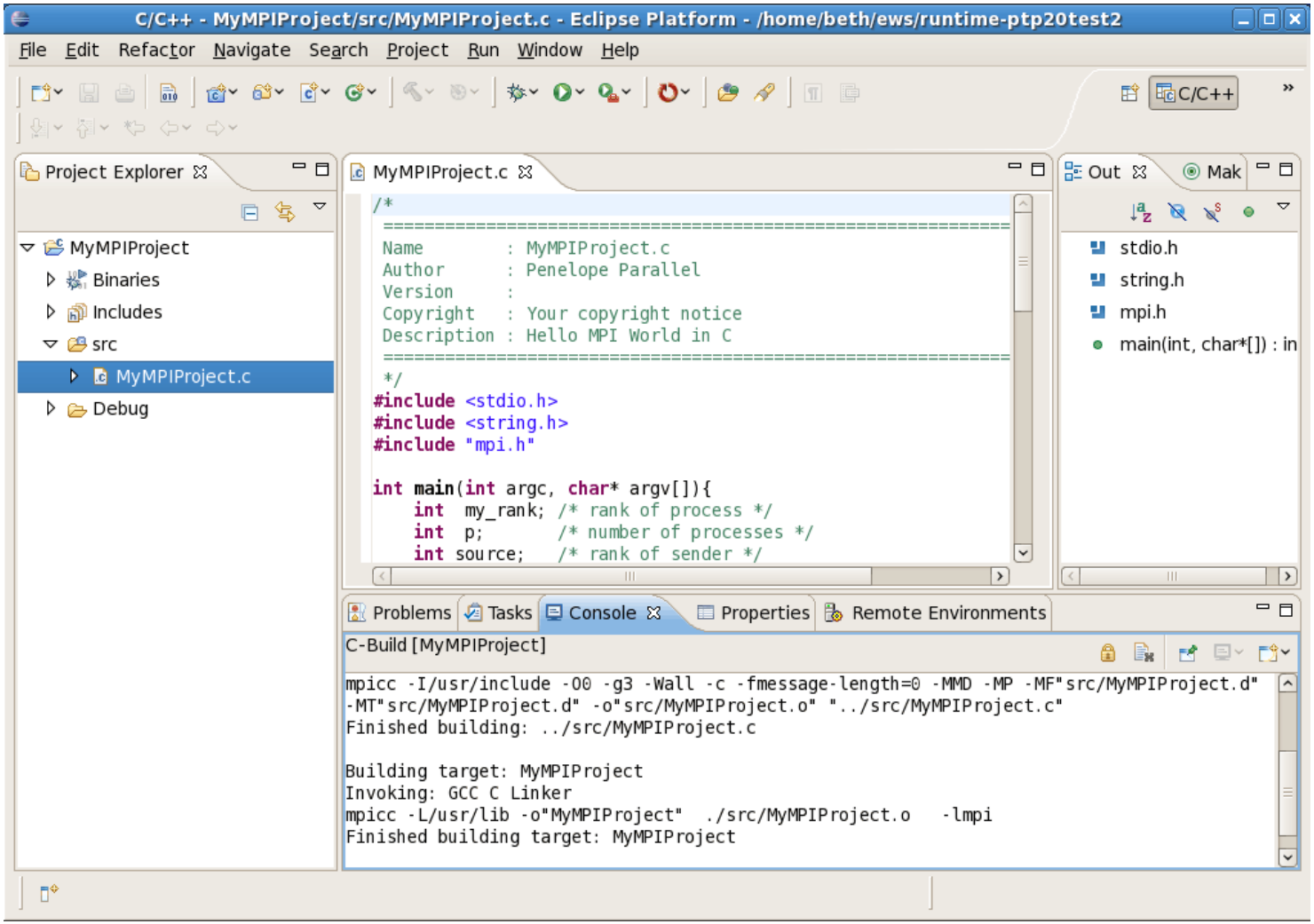
- Sử dụng hệ thống song song sẵn có
  - Không đòi hỏi cấu hình, bảo trì, thay đổi máy tính cá nhân
  - Tập trung vào viết chương trình
  - Số lượng các nút tính toán nhiều
  - Môi trường song song thực sự
  - Hệ thống bkluster
    - Địa chỉ: bkluster.hut.edu.vn
    - Liên hệ người quản trị để lấy tài khoản đăng nhập
- Tự cài đặt môi trường song song trên máy cá nhân
  - Số lượng CPU ít
  - Đòi hỏi công sức cài đặt, bảo trì

Các bước cài đặt

- Hướng dẫn này áp dụng cài trên 1 máy tính đơn
- Bước 1: Hệ điều hành
  - Cài đặt hệ điều hành dựa trên Linux: OpenSuse, Ubuntu, Fedora, Debian, …
  - Có thể dùng máy ảo để cài hệ điều hành
  - Hoặc cài đặt Cygwin trên môi trường Windows để thử nghiệm
- Bước 2: môi trường truyền thông điệp
  - Download phần mềm lammpi
    - http://www.lam-mpi.org/download/files/lam-7.1.4.tar.bz2
  - Giải nén
    - $ tar xjvf lam-7.1.4.tar.bz2
  - Cài đặt
    - Gõ lần lượt các lệnh sau: ./configure, make, make install

Môi trường soạn thảo mã nguồn

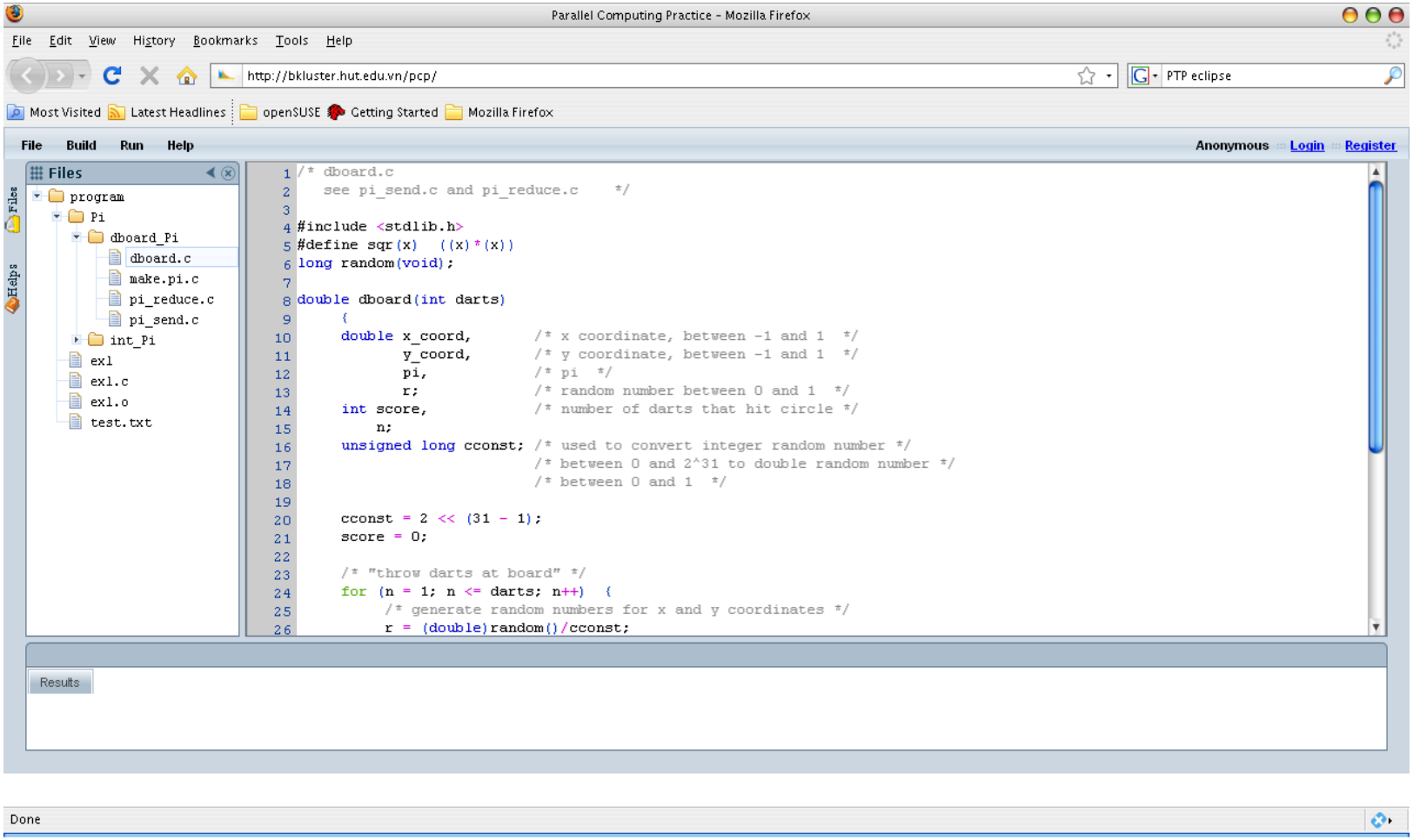
Chương trình eclipse, với plugin PTP



Môi trường soạn thảo mã nguồn

Môi trường lập trình song song trực tuyến PCP

http://bkluster.hut.edu.vn/pcp



Viết chương trình helloworld.c

```
1. #include <stdio.h>
2. #include <unistd.h>
3. #include <mpi.h>
4. int main(int argc, char* argv[])
5. {
6.         int size, rank;
7.         char  hostname[50];
8.         // Khoi tao tham so cho moi truong MPI
9.         MPI_Init(&argc, &argv);
10.        // Lay ve kich thuoc pommunicator
11.        MPI_Comm_size(MPI_COMM_WORLD, &size);
12.        // Lay ve so hieu rank
13.        MPI_Comm_rank(MPI_COMM_WORLD, &rank);
14.        // Lay ve hostname
15.        gethostname(hostname,50);
16.        printf("My rank is %d , %s\n",rank, hostname);
17.        MPI_Finalize();
18.        return 0;
19.}
```

Biên dịch chương trình

- Dùng lệnh
  - mpicc helloworld.c –o helloworld
  - Biên dịch mã nguồn ***helloworld.c*** thành chương trình chạy có tên helloworld

Khởi tạo môi trường LAMMPI

- Khởi động:
  - Lệnh: ***lamboot –v hostfile***
  - Tệp tin hostfile chưa tên các máy cần để chạy chương trình song song
    - Nếu hệ thống song song chỉ có một máy đơn, thì tệp tin này chỉ cần có 1 dòng có nội dung: localhost

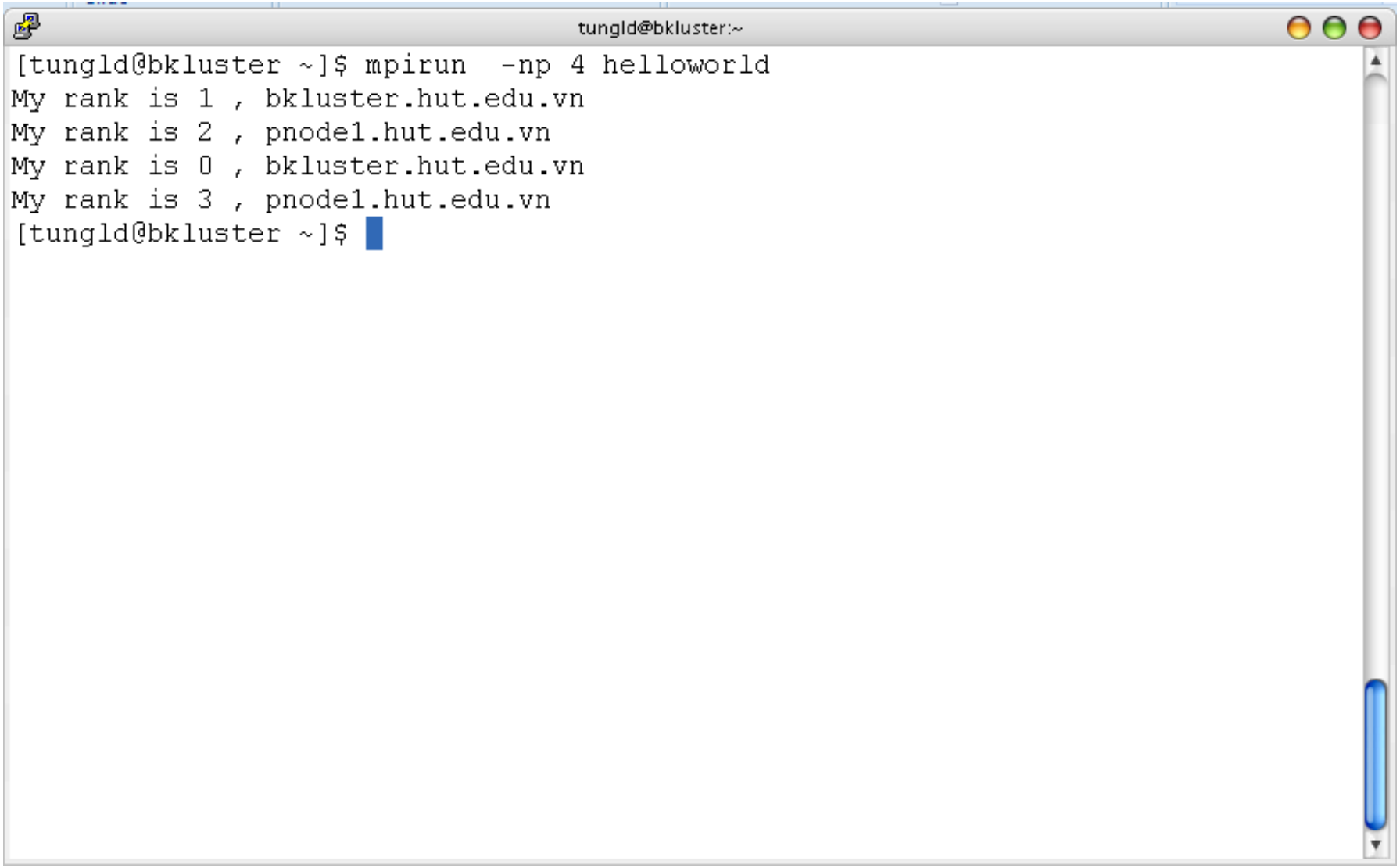
- Xem thông tin môi trường đã khởi động
  - lamnodes

```
# vi du hostfile
localhost cpu=2
```

Chạy chương trình helloworld

- Chạy chương trình
  - mpirun –np 4 helloworld
  - Chạy chương trình hello với 4 tiến trình
- Xem chương trình có thực sự chạy không:
  - Mở 1 terminal mới
  - Gõ lệnh: ps –aux | grep tenchuongtrinh
    - ps –aux | grep helloworld
- Tắt môi trường lam/mpi
  - lamhalt

Kết quả


tungld@bkluster:~
[tungld@bkluster ~]$ mpirun -np 4 helloworld
My rank is 1 , bkluster.hut.edu.vn
My rank is 2 , pnode1.hut.edu.vn
My rank is 0 , bkluster.hut.edu.vn
My rank is 3 , pnode1.hut.edu.vn
[tungld@bkluster ~]$

Demo

- Chạy chương trình helloworld trên hệ thống song song bkluster
- Đánh giá, nhận xét đặc tính chương trình song song