**高性能并行计算第5次作业**

姓名：姚虎成 学号：2020317110033

**代码地址：**

MPI hello world 的C语言程序路径：

/home2/2020317110033/mpi5/mpi\_hello.c

**实验结果：**

编译mpi\_hello.c并运行，输入./mpi\_hello,得到以下输出结果

Hello World from Node 0

I am 0 of 0

**图1 ./mpi\_hello 运行结果**

使用 mpirun −np 10 mpi\_hello 运行，得到图2结果

Hello World from Node 2

I am 2 of 0

Hello World from Node 8

I am 8 of 0

Hello World from Node 9

I am 9 of 0

Hello World from Node 0

I am 0 of 0

Hello World from Node 1

I am 1 of 0

Hello World from Node 3

I am 3 of 0

Hello World from Node 4

I am 4 of 0

Hello World from Node 5

I am 5 of 0

Hello World from Node 6

I am 6 of 0

Hello World from Node 7

I am 7 of 0

**图2 设置10个核 运行结果**

**实验分析：**

首先配置MPI程序时，使用source命令引用opt/openmpi.bashrc 目录下的配置文件

source /opt/openmpi.bashrc

配置完成后，可以使用man mpi 查看mpi的帮助命令。熟悉了mpi以及mpirun命令的基本参数，开始写MPI hello world 程序。

MPI hello world程序 需要头文件mpi.h 才能运行，首先需要初始化MPI环境，然后通过一些环境变量可以获得当前处理核心的编号和节点数。再通过MPI\_Finalize释放一些资源。

需要将mpi\_hello.cC语言程序编译成可执行程序mpi\_hello.

mpicc mpi\_hello.c -o mpi\_hello

实现多核心并设置核心数目的命令为

mpirun -np 10 mpi\_hello

本次练习通过编写mpi的hello world程序，熟悉了最基本的MPI程序执行过程，需要先使用MPI\_Init(&argc,&argv)初始化环境，执行计算后，也需要MPI\_Finalize来结束MPI环境

**附录：**

计算pi的C语言程序代码

#include <stdio.h>

#include <mpi.h>

main(int argc, char \*\*argv)

{

int node,numprocs;

MPI\_Init(&argc,&argv);

MPI\_Comm\_rank(MPI\_COMM\_WORLD, &node);

printf("Hello World from Node %d\n",node);

printf(" I am %d of %d \n",node,numprocs);

MPI\_Finalize();

}