

# MPI 编程

姓名：郁万祥 学号：2013852

环境配置（华为云）：

以成功运行 hello word 程序为例：

```
[yuwanxiang@ecs-hw-0001 hello]$ mpiexec -n 6 -f /home/yuwanxiang/hello/config /home/yuwanxiang/hello/mpi_hello_world
Authorized users only. All activities may be monitored and reported.
Authorized users only. All activities may be monitored and reported.
Hello world from processor ecs-hw-0003, rank 4 out of 6 processors
Hello world from processor ecs-hw-0002, rank 2 out of 6 processors
Hello world from processor ecs-hw-0003, rank 5 out of 6 processors
Hello world from processor ecs-hw-0002, rank 3 out of 6 processors
Hello world from processor ecs-hw-0001, rank 0 out of 6 processors
Hello world from processor ecs-hw-0001, rank 1 out of 6 processors
[yuwanxiang@ecs-hw-0001 hello]$ ^C
[yuwanxiang@ecs-hw-0001 hello]$
```

问题一、

实现第 5 章课件中的梯形积分法的 MPI 编程熟悉并掌握 MPI 编程方法，规模自行设定，可探讨不同规模对不同实现方式的影响。

源码见附录：[question1.c](#)

1、规模 2000:

```
[yuwanxiang@ecs-hw-0001 hello]$ mpiexec -n 6 -f /home/yuwanxiang/hello/config /home/yuwanxiang/hello/mpi_hello_world
Authorized users only. All activities may be monitored and reported.
Authorized users only. All activities may be monitored and reported.
total size: 333.000030
[yuwanxiang@ecs-hw-0001 hello]$ ^C
[yuwanxiang@ecs-hw-0001 hello]$ ^C
[yuwanxiang@ecs-hw-0001 hello]$
```

2、规模 4000:

```
[yuwanxiang@ecs-hw-0001 hello]$ mpiexec -n 6 -f /home/yuwanxiang/hello/config /home/yuwanxiang/hello/mpi_hello_world
Authorized users only. All activities may be monitored and reported.
Authorized users only. All activities may be monitored and reported.
total size: 333.000008
[yuwanxiang@ecs-hw-0001 hello]$ ^C
[yuwanxiang@ecs-hw-0001 hello]$
```

3、规模 6000:

```

mpicc -o mpi_hello_world mpi_hello_world.c
[yuwanxiang@ecs-hw-0001 hello]$ mpiexec -n 6 -f /home/yuwanxiang/hello/config /home/yuwanxiang/hello/mpi_hello_world

Authorized users only. All activities may be monitored and reported.

Authorized users only. All activities may be monitored and reported.
total size: 333.000003
[yuwanxiang@ecs-hw-0001 hello]$ ^C

```

结果分析：

规模	2000	4000	6000
结果	333.000030	333.000008	333.000003

我们可以发现，在积分过程中，规模越大，最终结果的准确度越高。

但是规模提升了一倍，准确率提高了 0.000022，这个数据是很小的，因此，我们是否需要为了微小的准确度而倍增规模，还需要根据实际问题确定。

问题二、

对于课件中“多个数组排序”的任务不均衡案例进行 MPI 编程实现，规模可自行设定、调整。

源码见附录：[question2.cpp](#)

### 1、规模 2048:

```

[yuwanxiang@ecs-hw-0001 ~]$ mpiexec -n 6 -f /home/yuwanxiang/hello/config /home/yuwanxiang/hello/mpi_hello_world
[yuwanxiang@ecs-hw-0001 hello]$ mpiexec -n 6 -f /home/yuwanxiang/hello/config /home/yuwanxiang/hello/mpi_hello_world

Authorized users only. All activities may be monitored and reported.

Authorized users only. All activities may be monitored and reported.
ARR_NUM = 2048
MPI_arraysort time: 0.000008s[yuwanxiang@ecs-hw-0001 hello]$ ^C

```

### 2、规模 4096:

```

[yuwanxiang@ecs-hw-0001 ~]$ mpiexec -n 6 -f /home/yuwanxiang/hello/config /home/yuwanxiang/hello/mpi_hello_world
[yuwanxiang@ecs-hw-0001 hello]$ mpiexec -n 6 -f /home/yuwanxiang/hello/config /home/yuwanxiang/hello/mpi_hello_world

Authorized users only. All activities may be monitored and reported.

Authorized users only. All activities may be monitored and reported.
ARR_NUM = 4096
MPI_arraysort time: 0.000018s[yuwanxiang@ecs-hw-0001 hello]$ ^C
[yuwanxiang@ecs-hw-0001 hello]$ 

```

### 3、规模 8192:

```
[yuwanxiang@ecs-hw-0001 hello]$ mpicxx mpi_hello_world.cpp -o mpi_hello_world
[yuwanxiang@ecs-hw-0001 hello]$ mpiexec -n 6 -f /home/yuwanxiang/hello/config /home/yuwanxiang/hello/mpi_hello_world

Authorized users only. All activities may be monitored and reported.

Authorized users only. All activities may be monitored and reported.
ARR_NUM = 8192
MPI_arraysort time: 0.000032s[yuwanxiang@ecs-hw-0001 hello]$
```

数组规模	2048	4096	8192
运行时间	0.000008	0.000018	0.000032

很显然，随着数组规模增大，运行耗时随着理应增加。

附加题、

实现高斯消去法解线性方程组的 MPI 编程，与 SSE(或 AVX)编程结合，并与 Pthread、OpenMP（结合 SSE 或 AVX）版本对比，规模自己设定。

源码见附录：[question3.cpp](#)

### 1、规模 1024:

```
MPIDB Complete posted with error(1137): Process failed
[yuwanxiang@ecs-hw-0001 hello]$ mpiexec -n 6 -f /home/yuwanxiang/hello/config /home/yuwanxiang/hello/mpi_hello_world

Authorized users only. All activities may be monitored and reported.

Authorized users only. All activities may be monitored and reported.
size : 1024
all work done!!!!
time: 381.861000 ms
```

### 2、规模 2048:

```
[yuwanxiang@ecs-hw-0001 hello]$ mpicxx mpi_hello_world.cpp -o mpi_hello_world
[yuwanxiang@ecs-hw-0001 hello]$ mpiexec -n 6 -f /home/yuwanxiang/hello/config /home/yuwanxiang/hello/mpi_hello_world

Authorized users only. All activities may be monitored and reported.

Authorized users only. All activities may be monitored and reported.
size : 2048
all work done!!!!
time: 655.648000 ms
```

### 3、规模 4096:

```
[yuwanxiang@ecs-hw-0001 hello]$ mpicxx mpi_hello_world.cpp -o mpi_hello_world
[yuwanxiang@ecs-hw-0001 hello]$ mpiexec -n 6 -f /home/yuwanxiang/hello/config /home/yuwanxiang/hello/mpi_hello_world

Authorized users only. All activities may be monitored and reported.

Authorized users only. All activities may be monitored and reported.
size : 4096
all work done!!!!
time: 950.099000 ms
```

#### MPI 与 openMP、Pthread 的比较:

首先 MPI 是基于分布式内存系统，而 openMP 和 Pthread 是基于共享内存系统；也就是说 MPI 之间的数据共享需要通过消息传递，因为 MPI 同步的程序属于不同的进程，甚至不同的主机上的不同进程。相反由于 openMP 和 Pthread 共享内存，不同线程之间的数据就无须传递，直接传送指针就行。

同时 MPI 不同主机之间的进程协调工作需要安装 MPI 软件来完成。

在 openMP 和 Pthread 之间的区别主要在编译的方式上，openMP 的编译需要添加编译器预处理指令 `#pragma`，创建线程等后续工作要编译器来完成。而 Pthread 就是一个库，所有的并行线程创建都需要我们自己完成。