

# Homework 4

PB17000297 罗晏宸

May 24 2020

## 1 Exercise 15.1

```
process 0:
MPI_Send(msg1, count1, MPI_INT, tag1, comm1);
parallel_fft(...);
process 1:
MPI_Send(msg1, count1, MPI_INT, tag1, comm1);
parallel_fft(...);
```

- ① 试分析上述代码段的计算功能
- ② 如果在 `parallel_fft(...)` 中又包含了另一个发送程序：

```
if (my_rank == 0) MPI_Send(msg2, count1,
    MPI_INT, 1, tag2, comm2);
```

如果没有通信体则会发生什么情况？

解

① 代码段中的 `MPI_Send` 函数参数列表中省略了 `int dest` 参数，认为进程 0 将起始地址为 `msg1` 的 `count1` 个整型数据加上标签 `tag1` 发送给通信域 `comm1` 中的目标进程 1，进程 1 将相同类型的消息缓冲发送给同一通信域中的目标进程 0。考虑到 `parallel_fft(...)` 中的消息接收，代码通过两个进程实现了快速傅里叶变换的并行计算。

② 没有通信体则程序无法区分在通信域 `comm1` 和 `comm2` 中编号同为 1 的两个进程，发送给通信域 `comm1` 中的目标进程 1 的消息 `msg1` 会与 `msg2` 相互干扰，无法安全地区别不同的通信。

## 2 Exercise 15.3

填写空白处，使下述两代码段完全等效：

①

```
float data[1024];
MPI_Datatype floatttype;
MPI_Type_vector(10, 1, 32, MPI_FLOAT, &floatttype);
MPI_Type_commit(&floatttype);
MPI_Send(data, 1, floatttype, dest, tag,
         MPI_COMM_WORLD);
MPI_Type_free(&floatttype);
```

②

```
float data[1024], buff[10];
for(____; ____; i++) buff[i] = data[____];
MPI_Send(buff, ____, MPI_FLOAT, ____, ____, ____);
```

解

②

```
float data[1024], buff[10];
for(int i = 0; i < 10; i++)
    buff[i] = data[32 * i];
MPI_Send(buff, 10, MPI_FLOAT, dest, tag,
         MPI_COMM_WORLD);
```

### 3 Exercise 15.13

(Buffon-Laplace 针问题) 设想一个长为  $l$  的针掉在一个等距平行线网格上, 每个格的长和宽分别是  $a$  和  $b$ 。针至少落在一根线上的概率为

$$P(l, a, b) = \frac{2l(a+b) - l^2}{\pi ab}$$

我们可以用蒙特卡洛模拟法进行投针, 从而来估计  $\pi$  的值

① 用 C 语言写一个串行的 Buffon-Laplace 针问题的仿真程序。程序打印  $\pi$  的值。当针的数量是一百万时, 运行模拟的时间是什么?  $\pi$  的位数是多少?

解

① 详细程序可见附件, 程序中使用了如下函数模拟统计投针落在线上的次数

```
int BuffonLaplaceSimulation(double l, double a,
    double b, int n)
{
    int hits = 0;
    double x1, y1, x2, y2;
    double angle;
    for (int i = 0; i < n; i++)
    {
        x1 = a * (double)rand() /
            (double)RAND_MAX;
        y1 = b * (double)rand() /
            (double)RAND_MAX;
        angle = 2.0 * M_PI * (double)rand() /
            (double)RAND_MAX;
        x2 = x1 + l * cos(angle);
        y2 = y1 + l * sin(angle);
        if (x2 <= 0 || x2 >= a || y2 <= 0 || y2
            >= b)
            hits++;
    }
}
```

```
        return hits;  
    }
```

当取  $l = a = b = 2$  时，进行一百万次投针程序运行时间约为0.183s，计算得到  $\pi$  结果精确到小数点后 2 位，更改  $l, a, b$  的值，程序运行时间没有显著变化，结果精确位数没有改变。