

# HOMEWORK 2

## 作业内容

使用MPI实现附件PDF文件中 2.7.1 节描述的直方图程序，进程 0 读取输入的数据，并将它们分配到其余进程，最后进程 0 打印该直方图。

请根据提供的代码，将代码补充完整！需要补充的部分见注释 **PLEASE ADD THE RIGHT CODES** 部分。（详细内容请下载附件 **mpi\_histogram.zip**查看

## 作业要求

1. 提交作业的时请将修改好的 mpi\_histogram.c 文件压缩为zip文件后上传
2. DDL： 3月28日23:59前
3. 禁止抄袭

## 解答过程

### 问题分析

首先明确需求，我们需要设计一个并行程序，这个程序的进程0读取输入的数据，并将它们分配到其余进程，最后进程 0 打印该直方图。

根据题目，输入的数据有：

- 1) 数据的个数data\_count
- 2) 长度为data\_count的浮点数数组data
- 3) 数据的最小值min\_meas
- 4) 数据的最大值max\_meas
- 5) 桶的个数bin\_count

### 进程0读入数据

```
1 void Get_input(int* bin_count_p, float* min_meas_p, float* max_meas_p,  
2     int* data_count_p, int* local_data_count_p, int my_rank,  
3     int comm_sz, MPI_Comm comm);  
4 void Check_for_error(int local_ok, char fname[], char message[],  
5     MPI_Comm comm);  
6 void Gen_data(float local_data[], int local_data_count, int data_count,  
7     float min_meas, float max_meas, int my_rank, MPI_Comm comm);
```

在本题中，在Get\_input中进程0读入数据的个数data\_count、数据的最大值max\_meas、数据的最小值min\_meas以及桶的个数bin\_count，同时将这些参数广播给各个进程，data由输入的数据最大值和数据最小值以及数据个数随机生成（Gen\_data），Check\_for\_error检查数据的个数是否能被进程数整除，若不能整除则退出。

## 将数据分配给各个进程

```
1 void Set_bins(float bin_maxes[], int loc_bin_cts[], float min_meas,  
2             float max_meas, int bin_count, int my_rank, MPI_Comm comm);
```

Set\_bins的输出为bin\_maxes和loc\_bin\_cts，由进程0完成，它根据数据的最小值min\_meas和数据的最大值max\_meas以及桶的个数bin\_count确定每个区间的范围，定义每个桶的区间长度bin\_width

$$bin\_width = \frac{(max\_meas - min\_meas)}{bin\_count}$$

获取bin\_maxes的方法：

```
1 for (b = 0; b < bin_count; b++)  
2     bin_maxes[b] = min_meas + bin_width*(b+1);
```

同时将loc\_bin\_cts各元素初始化为0。

## 各个进程执行

```
1 void Find_bins(int bin_counts[], float local_data[], int loc_bin_cts[],  
2               int local_data_count, float bin_maxes[],  
3               int bin_count, float min_meas, MPI_Comm comm);  
4 int which_bin(float data, float bin_maxes[], int bin_count,  
5             float min_meas);
```

Find\_bins将各个数据入桶，实现方法如下：

```
1 for (i = 0; i < local_data_count; i++){  
2     bin = which_bin(local_data[i], bin_maxes, bin_count, min_meas);  
3     loc_bin_cts[bin]++;  
4 }
```

调用了函数Which\_bin，Which\_bin返回data应属于的桶的序号b，满足以下条件

$$bin\_maxes[b - 1] \leq measurement < bin\_maxes[b]$$

其中小于bin\_maxes[0]的数据将被放入第一个桶，即有

$$min\_meas \leq measurement < bin\_maxes[0]$$

因此Which\_bin的实现方法如下：

```
1     int i = 0;  
2  
3     //The first bin  
4     if (data >= min_meas && data < bin_maxes[0])  
5         return i;  
6  
7     //Get the right bin  
8     while (i < bin_count && data >= bin_maxes[i] && data >= min_meas)  
9     {  
10         ++i;  
11     }  
12  
13     //Return the result
```

```

14         if (i < bin_count)
15             return i;
16

```

## 将各个进程的结果汇总

各个数据入桶后，需要将各个进程的结果汇总，实现方法如下：

```

1     MPI_Reduce(&loc_bin_cts, bin_counts, bin_count, MPI_FLOAT, MPI_SUM, 0,
    comm);

```

MPI\_Reduce是mpi库里的全局规约函数，它将所有的发送信息进行同一个操作。

```

int MPI_Reduce(
    void *input_data,           /* 指向发送消息的内存块的指针*/
    void *output_data,         /* 指向接收（输出）消息的内存块的指针*/
    int count,                  /* 数据量*/
    MPI_Datatype datatype,     /* 数据类型*/
    MPI_Op operator,           /* 规约操作*/
    int dest,                   /* 要接收（输出）消息的进程的进程号*/
    MPI_Comm comm);            /* 通信器，指定通信范围*/

```

发送消息的内存块为loc\_bin\_cts，接收（输出）消息的内存块为bin\_counts，数据量为bin\_count，数据类型为MPI\_FLOAT，规约操作为MPI\_SUM，规约到进程0，因此要接收（输出）消息的进程的进程号为0，通信子为当前通信子。

## 运行结果

输入为20个0~5的随机数，桶的数量为5：

```
5 0 5 20
```

跑4个进程，运行结果如下：

运行结果 ☺

```

===== ERROR =====

===== OUTPUT =====
Enter the number of bins
Enter the minimum measurement
Enter the maximum measurement
Enter the number of data
Generated data:
0.612 4.710 4.623 3.630 1.275 2.973 0.740 3.947 1.498 2.940 2.246 1.450 0.653 4.515 1.706 2.957 0.784 3.887 1.291 2.515

0.000-1.000: XXXX
1.000-2.000: XXXXX
2.000-3.000: XXXXX
3.000-4.000: XXX
4.000-5.000: XXX

===== REPORT =====

```

结果符合预期，第一个桶有4个数，分别为0.612、0.740、0.653、0.784；

第二个桶有5个数，分别为1.275、1.498、1.450、1.706、1.291；

第三个桶有5个数，分别为2.973、2.940、2.246、2.957、2.515；

第四个桶有3个数，分别为3.630、3.947、3.887；

第五个桶有3个数，分别为4.710、4.623、4.515。

结果符合预期。

当随机数的数量为60时，结果如下：

运行结果

```
===== ERROR =====

===== OUTPUT =====
Enter the number of bins
Enter the minimum measurement
Enter the maximum measurement
Enter the number of data
Generated data:
0.612 4.710 4.623 3.630 1.275 2.973 0.740 3.947 1.498 2.940 2.246 1.450 0.653 4.515 1.706 2.957 0.784 3.887 1.291 2.515 2.779 4.695 2.595 4.925 4.943 2.288 1.048 1.023 4.013 4.812 0.002 4.625
4.522 4.625 3.255 0.797 2.599 3.995 4.744 4.097 1.935 1.990 0.546 2.588 1.505 2.252 0.545 2.289 1.139 1.836 4.804 3.918 1.531 2.399 3.843 1.474 4.686 4.891 2.497 3.699

0.000-1.000: XXXXXXXX
1.000-2.000: XXXXXXXXXXXXXX
2.000-3.000: XXXXXXXXXXXXXXXX
3.000-4.000: XXXXXXXX
4.000-5.000: XXXXXXXXXXXXXXXX

===== REPORT =====
```

经分析，结果符合预期。