# 高性能计算导论 第三次作业

（中英文题意不一致时，以中文的要求为准！）

## Exercise 3.13

**MPI\_Scatter** and **MPI\_Gather** have the limitation that each process must send or receive the same number of data items. When this is not the case, we must use the MPI functions **MPI\_Gatherv** and **MPI\_Scatterv**. Look at the man pages for these functions, and modify your vector sum program so that it can correctly handle the case when **n** isn’t evenly divisible by **comm\_sz**.

修改课本附带的ch3/mpi\_vector\_add.c程序，利用**MPI\_Gatherv** and **MPI\_Scatterv**函数使得它可以正确处理n不被进程数整除的情况。然后对程序编译、运行，简单测试是否达到了效果。

## Exercise 3.11(d)

Finding **prefix sums** is a generalization of global sum. Rather than simply finding the sum of n values,

the prefix sums are the n partial sums,

1. MPI provides a collective communication function, **MPI\_Scan**, that can be used to compute prefix sums:



It operates on arrays with **count** elements; both **sendbuf\_p** and **recvbuf\_p** should refer to blocks of count elements of type **datatype**. The **op** argument is the same as **op** for **MPI\_Reduce**. Write an MPI program that generates a random array of **count** elements on each MPI process, finds the prefix sums, and prints the results.

计算前缀和是求和问题的推广，与求n个数和：

不同，n个数值的前缀和(prefix sum)是如下n个部分和，

1. MPI提供一个集合通信函数MPI\_Scan，用来计算前缀和（第i个进程接收第i个前缀和的结果）：



其中，sendbuf\_p和recvbuf\_p都应该指向有count个datatype类型元素的数据块，op参数和MPI\_Reduce中的op一样（这里应是MPI\_SUM）。编写一个MPI程序，使每个进程生成有10个元素的随机数组，共有p个进程，10p个元素的全局数组（按0号、1号、2号… 进程顺序排列）。计算这个全局数组的10p个前缀和, 存在p个进程中。最后，将前缀和传输给0号进程，打印输出。

（提示：先在各个进程算局部前缀和，然后再利用MPI\_Scan函数处理局部和）

## Programming Assignments 3.1

Use MPI to implement the histogram program discussed in Section2.7.1. Have process 0 read in the input data and distribute it among the processes. Also have process 0 print out the histogram.

使用MPI实现2.7.1节讨论的直方图程序，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。

## Programming Assignments 3.5

Implement matrix-vector multiplication using a block-column distribution of the matrix. You can have process 0 read in the matrix and simply use a loop of sends to distribute it among the processes. Assume the matrix is square of order n and that n is evenly divisible by **comm\_sz**. You may want to look at the MPI function **MPI\_Reduce\_scatter**.

实现矩阵与向量相乘的程序，其中矩阵以列为单位分块。进程0读取输入的矩阵，然后循环地向其他进程分发数据。假定矩阵是矩阵，n能够被comm\_sz整除。你可能需要下查看一下MPI函数MPI\_Reduce\_scatter的使用方法。

测试不同的n值和不同的p值（即comm\_sz）时的加速比和并行效率。例如n=20000, 2000, p=1, 2, 4, 8.

## Programming Assignments 3.6

Implement matrix-vector multiplication using a block-submatrix distribution of the matrix. Once again, you can have process 0 read in the matrix and aggregate the sub-matrices before sending them to the processes. Assume comm\_sz is a perfect square and that evenly divides the order of the matrix.

实现矩阵与向量相乘的程序，其中矩阵以子矩阵块为单位分割。同样，由进程0读取整个矩阵，将矩阵分块为子矩阵后，发送给其余进程。假设comm\_sz是个完全平方数，且能够整除矩阵的阶n。

测试不同的n值和不同的p值（即comm\_sz）时的加速比和并行效率。例如n=20000, 2000, p=1, 4, 16, 25.

## 作业要求

1. 每道题都要编程，完成的作业分“答题文档”和“程序附件”两部分，打包压缩后提交。在答题文档中，对各题简略介绍解题思路，贴出关键程序和测试结果。
2. 编程作业3.1提供了程序prog3.1\_histo\_dist.c，请补充完整其中的函数Find\_bins和它调用的Which\_bin函数。
3. 编程作业3.5, 3.6需基于prog3\_56.cpp程序完成C++语言程序，编译使用make，它使用的编译规则见Makefile文件。或者，基于附上的prog3.5\_mpi\_mat\_vect\_col.c程序完成3.5题、3.6题程序的编写，但要求修改程序使得输入的矩阵向量为随机矩阵、向量。
4. 编程作业3.5、3.6的附加说明：
   1. 所有的矩阵与向量均由主进程生成或者读入，然后分发到其他进程中去；
   2. 请精心设计实验n、p值，以满足题目中整除与完全平方数的条件。原题中的n与p值仅为示例，可以使用不同的矩阵规模与进程数目。例如更大的矩阵规模和更多的进程数目；
   3. 并行结果的正确性检验是必须的，请计算并行计算结果与串行计算结果的差（采用向量2范数度量），并输出；
   4. 请分别统计不计算数据分发时间与计算数据分发时间的耗时与加速比；
   5. 答题文档中应当包含：不同n与p值的数据分发时间、计算时间、总时间、加速比的表格及折线图
   6. 请完成Makefile文件，以保证助教使用make后可以在集群上正常编译出运行程序。可以添加make run提供一个你的可执行程序的包含参数配置的运行样例，或在文档中说明你的参数含义。