

MAP55611 HPC Software Assignment 1

1 Notes

- To complete these exercises, submit a document containing results for each question and a separate tarball of the all code used to obtain the results. Details of how to compile and run the code must be provided in a “README” file. You must use a makefile to compile the code. The code must compile on seagull.

为完成这些练习，请提交一个包含每个问题结果的文档和一个包含所有用于获取结果的代码的单独压缩包。如何编译和运行代码的详细信息必须在一个“README”文件中提供。您必须使用 makefile 来编译代码。代码必须能在 seagull 上编译。

- Use C to implement all exercises
使用 C 语言实现所有练习。

2 Exercises

1. Read vector from file. Write code to:

从文件读取向量。编写代码以：

- read a vector from file on rank 0. The file format should be:
在 rank 0 上从文件读取一个向量。文件格式应该为：
 - length of the vector on the first line
第一行为向量的长度
 - entries of the vector, separated by spaces or carriage returns, on the remaining line or lines
向量的条目，用空格或回车分隔，在剩余的行中
- Evenly divide the vector among the processes (you can assume that the number of processors divides the length of the vector). On each processor you should only allocate enough memory for the local subset of the vector assigned to that processor
在进程之间平均划分向量（您可以假设处理器的数量可以整除向量的长度）。在每个处理器上，您应该只为分配给该处理器的向量的本地子集分配足够的内存
- Add 1.0 to each element of the local vector on each processor
在每个处理器上给本地向量的每个元素加上 1.0
- gather the local vectors back into a vector on rank 0 and write this vector to a file of the same format as the input file
将本地向量重新收集到 rank 0 上的一个向量中，并将此向量写入与输入文件相同格式的文件中
- Run this code on 4 processors and for a vector of length $n = 16$. Run the code on at least 10 processors for a vector length at $n = 10000$
在 4 个处理器上运行此代码，并且向量长度为 $n = 16$ 。在至少 10 个处理器上运行代码，向量长度为 $n = 10000$
- Provide two implementations of this code: one which uses MPI_Send / MPI_Recv and a second which uses MPI collective calls
提供此代码的两种实现：一种使用 MPI_Send / MPI_Recv，另一种使用 MPI 集合调用

2. One dimensional decomposition of a list of items of length n over p processors. Write a function to decompose an array of doubles of size n across p processors. The code should return for each process the local starting point into the global array and the local end point into the global array. You cannot assume that p divides n without remainder. The interface in C must be

```
int decomp1d(int n, int p, int myid, int *s, int *e)
```

where $myid$ is the rank of the processor; n , p , $myid$ are inputs and s and e are outputs. The return value should be 0 on success.

对长度为 n 的项目列表在 p 个处理器上进行一维分解。编写一个函数来分解大小为 n 的双精度数组跨 p 个处理器。代码应为每个进程返回进入全局数组的本地起点和进入全局数组的本地终点。您不能假设 p 可以无余数地除以 n 。C 语言中的接口必须是

```
int decomp1d(int n, int p, int myid, int *s,int *e)
```

其中 $myid$ 是处理器的排名; n 、 p 、 $myid$ 是输入, s 和 e 是输出。成功时返回值应为 0。

- Ensure that the decomposition is *load balanced* – numbers of elements assigned to different processes should differ by at most one

确保分解是 *load balanced* - 分配给不同进程的元素数量最多相差一个

- Note that this function does not need to contain any MPI code –it just returns values for s and e based on values of n , p , $myid$

注意, 此函数不需要包含任何 MPI 代码 - 它只是根据 n 、 p 、 $myid$ 的值返回 s 和 e 的值

- Use your function in a basic MPI code which calls the decomposition function and prints out the results for each rank. Demonstrate that the decomposition produces consistent results on up to 9 processors for $n = 25$. Do the same on 10 processors for $n = 100000$. You do not need to allocate any memory for the vector, just print the output of the function

在一个基本的 MPI 代码中使用您的函数, 该代码调用分解函数并打印出每个等级的结果。证明该分解在最多 9 个处理器上产生一致的结果, $n = 25$ 。在 10 个处理器上对 $n = 100000$ 做同样的事情。您不需要为向量分配任何内存, 只需打印函数的输出

- Demonstrate that the function also produces correct results when run on *one processor*

证明当在 *one processor* 上运行时, 该函数也能产生正确的结果

3. Read a vector from file and calculate the square of the Euclidean (l_2) norm of the vector ($\|x\|_2^2$ = the sum of squared entries of the vector) Using a similar file format used in Q1 (see the attached files `q3file_16.txt` and `q3file_97.txt`)

从文件中读取向量并计算向量的欧几里得 (l_2) 范数的平方 ($\|x\|_2^2$ = 向量中各项的平方和)。使用与 Q1 中类似的文件格式 (参见附加文件 `q3file_16.txt` 和 `q3file_97.txt`)

- read a vector from file on rank 0 在 rank 0 上从文件读取一个向量
- distribute the vector among p processors. You cannot assume the number of processors divides the length of the vector without remainder. Use the function from Q2 or otherwise to divide the vector among the processors in a load balanced way. Different processors may have different local vector sizes

在 p 个处理器之间分配向量。不能假设处理器的数量可以无余数地划分向量的长度。使用 Q2 中的函数或其他方法以负载均衡的方式在处理器之间划分向量。不同的处理器可能有不同的本地向量大小

- on each processor calculate $\|x_{local}\|_2^2$
在每个处理器上计算 $\|x_{local}\|_2^2$
- calculate the full sum of squares by combining all local results onto rank 2 and print the answer
通过将所有本地结果合并到 rank 2 上并打印答案, 计算平方和的总和
- calculate the full sum of squares by combining all local results onto all processes and print the answer

通过将所有本地结果合并到所有进程上并打印答案, 计算平方和的总和

- Using the files `q3file_16.txt` and `q3file_97.txt`, demonstrate that your answers obtained using the parallel code are correct when run on:

使用文件 `q3file_16.txt` 和 `q3file_97.txt`, 展示您使用并行代码运行时获得的答案是正确的, 运行条件包括:

- `q3file_16.txt` for 4 processors
对于 4 个处理器使用 `q3file_16.txt`
- `q3file_16.txt` for 9 processors
对于 9 个处理器使用 `q3file_16.txt`
- `q3file_16.txt` for 1 processor
对于 1 个处理器使用 `q3file_16.txt`
- `q3file_97.txt` for 11 processors
对于 11 个处理器使用 `q3file_97.txt`
- `q3file_97.txt` for 6 processors
对于 6 个处理器使用 `q3file_97.txt`