作业报告

操作流程：

1. 创建一个大数组char a[]，长度为100M×线程数，分别使用MPI\_Send()从0号线程发送并使用MPI\_Recv()接口从其他线程接收，使得每个线程保存100M大小的char arr[]数组。
2. 并行赋值。每个线程中将100M大小的字符串数组填满线程号。如第2号线程赋值完毕后将有100M-1个’2’和一个换行符’\n’。
3. 使用MPI\_File\_open()、MPI\_File\_set\_view()、MPI\_File\_write()、MPI\_File\_close()接口并行（按照线程号顺序）向文件data.out中写入刚才赋值的字符串数组。
4. 将各个进程大小为100M的char arr[]数组使用MPI\_send()发送回0号进程，并放在各自的位置中。
5. 第0号进程中检查接收的内容是否顺序且正确（即和文件中一致）。

运行记录：

　　使用mpirun program运行程序，在台式机上Intel(R) Core(TM) i9-9900K CPU @ 3.60GHz（8核）上运行输出如下。所有流程包括文件写入在HDD7200转硬盘上不到1秒运行时间。

Proc 1: Recieving from proc 0

Proc 5: Recieving from proc 0

Proc 6: Recieving from proc 0

Proc 7: Recieving from proc 0

Proc 3: Recieving from proc 0

Proc 4: Recieving from proc 0

Proc 2: Recieving from proc 0

1. Sending an array among processes evenly.

Copy from proc 0 to 0

Sending from proc 0 to 1

Sending from proc 0 to 2

Sending from proc 0 to 3

Sending from proc 0 to 4

Sending from proc 0 to 5

Sending from proc 0 to 6

Sending from proc 0 to 7

2. Parallel assigning values for every sub-arrays.

3. Parallel IO, writing to data.out file.

Wrote 100.000 MB data from rank 2 process

Sending from process 2 to 0

Wrote 100.000 MB data from rank 4 process

Sending from process 4 to 0

Wrote 100.000 MB data from rank 7 process

Sending from process 7 to 0

Wrote 100.000 MB data from rank 3 process

Sending from process 3 to 0

Wrote 100.000 MB data from rank 1 process

Sending from process 1 to 0

Wrote 100.000 MB data from rank 5 process

Sending from process 5 to 0

Wrote 100.000 MB data from rank 6 process

Sending from process 6 to 0

Wrote 100.000 MB data from rank 0 process

4. Sending sub-arraies to proc 0 and check.

Receiving 104857600 entries from process 1, with tag 0, error 0

Receiving 104857600 entries from process 2, with tag 0, error 0

Receiving 104857600 entries from process 3, with tag 0, error 0

Receiving 104857600 entries from process 4, with tag 0, error 0

Receiving 104857600 entries from process 5, with tag 0, error 0

Receiving 104857600 entries from process 6, with tag 0, error 0

Receiving 104857600 entries from process 7, with tag 0, error 0

Checking on process 0...

Conclusion: Array has same content with file!

# MPI

用MPI编写一个并行I/O的例子。

## 需求说明

### 创建一个1维数组

### 将1维数组在n个进程间均匀分布

可以采用：

连续分块方式

条块化分块方式

### 根据分块方案定义相应的filetype和etype

### 利用并行I/O接口把数据写入文件中

### 执行一个集合操作，把所有进程的分块数据发送给0号进程

可

采用文件I/O预定义的数据类型filetype或etype，

或者单独自定义数据类型，

或者直接使用基本数据类型，自己组织正确的数据布局，

### 0号进程另外开一个1维数组空间a用户保存这些收取的数据

接收的数据在a中的布局应保证跟文件/数组的逻辑布局一致

### 0号进程另起一个串行I/O读入并行I/O操作写入文件的数据

使用普通的文件读写

### 0号进程串行读入的输入放在1维数组空间b中

### 对比a和b验证I/O和通信是否正确

### 给出结论