

并行程序设计 with 算法

Assignment 2

2025-03-22

1: 简答题

Problem 1.1:

判断叙述的对错，并解释原因：

在 MPI 的消息传递场景下，对于同一个接收进程收到的来自任意个发送进程的多条消息，先发送的消息必须先到达。

Problem 1.2:

课本中，程序 3-2 调用的 Trap（程序 3-3）函数修改为如下实现，以输出中间结果便于调试算法。现在假设使用了 10 个进程进行计算，运行过程中会出现什么问题？

试给出问题的成因以及解决思路。

```
1 double Trap(  
2     double left_endpt /* in */,  
3     double right_endpt /* in */,  
4     int trap_count /* in */,  
5     double base_len /* in */) {  
6     double estimate, x;  
7     int i;  
8  
9     printf("Start calculating... --- ");  
10  
11     estimate = (f(left_endpt) + f(right_endpt))/2.0;  
12     for (i = 1; i <= trap_count-1; i++) {  
13         x = left_endpt + i*base_len;  
14         estimate += f(x);  
15     }  
16     estimate = estimate*base_len;  
17  
18     printf("Current estimate: %.6lf\n", estimate);  
19  
20     return estimate;  
21 } /* Trap */
```

Problem 1.3:

为什么 MPI 需要通信子？在使用 MPI_Bcast、MPI_Scatter、MPI_Gather 这样的集合通信函数时，如果通信子中只包含一个进程，这些集合通信函数分别会对输入输出缓冲区做什么操作？

2: 研究排序算法

排序是算法设计中十分经典的问题。以下是两种时间复杂度为 $O(n^2)$ 的不同风格算法实现。分析代码并回答以下问题。

```
1 def sort1(array, n):
2     for list_length in range(n, 1, -1):
3         for i in range(list_length - 1):
4             if array[i] > array[i+1]:
5                 temp = array[i]
6                 array[i] = array[i+1]
7                 array[i+1] = temp
8
9
10 def sort2(array, n):
11     for phase in range(n):
12         if phase % 2 == 0:
13             for i in range(1, n, 2):
14                 if array[i-1] > array[i]:
15                     temp = array[i]
16                     array[i] = array[i-1]
17                     array[i-1] = temp
18         else:
19             for i in range(1, n - 1, 2):
20                 if array[i] > array[i+1]:
21                     temp = array[i]
22                     array[i] = array[i+1]
23                     array[i+1] = temp
```

Problem 2.1:

这两个算法中，哪一个更容易充分并行化？若使用 p 个进程将其并行化，分析其最小的通信次数复杂度。（假设每次通信过程中，每两个进程可以互相交换自身拥有的全部数据。）

Problem 2.2:

各进程之间会相互通信。假设单个进程的通信策略如下所示。它们的通信是安全的还是不安全的？如果不安全，请给出一种策略解决它。

```
1 MPI_Send(msg, size, MPI_INT, (my_rank+1) % comm_sz, 0, comm);  
2 MPI_Recv(new_msg, size, MPI_INT, (my_rank+comm_sz-1) % comm_sz, -0,  
comm, MPI_STATUS_IGNORE);
```

Problem 2.3:

假设 n 和 p 都足够大。现在想要对算法进行测速。应该使用 CPU 时间还是墙上时间来测量自己编写的算法优劣？

为了公平起见，是否应该多次测量取平均值？

简述原因。

3: 模拟题

假定 $\text{comm_size} = 8$ ，向量 $x = (0, 1, 2 \dots 15)$ ($n=16$)。画图模拟以下过程：

Problem 3.1:

进程 0 使用树形结构通信分发到各进程，MPI_Scatter 的实现。

Problem 3.2:

通过块划分方式分配 x 给各个进程，使用蝶形通信结构实现聚集 x 的过程。