MPI 消息传递编程试题 (90 分钟完成)

2005 年 7 月 29 日

姓名_____ 单位_____

学号_____ (老师填写) 成绩_____

一、 填空题

(1) 任何 MPI 并行程序中都必须调用的 MPI 函数是_____ 和_____。**(3 分)**

(2) MPI 系统可直接用于通信的初始通信器有_____ 和_____。**(4 分)**

(3) MPI 的 C 语言接口中函数_____ 和_____ 的返回值不是 `int`。**(4 分)**

(4) 在二维 Poisson 方程的程序实例中，假设当前子区域不包含计算区域的边界，它所负责计算的网格点数为 $n \times m$，则每步迭代中所发送的总数据量为_____ 字节, 接收的总数据量为_____ 字节 (注：只考虑子区域边界间的通信，不考虑计算误差时 `MPI_Reduce` 的通信，并且假设实数长度为 4 个字节)。**(7 分)**

(5) 采用上道题中的假定。假设每个 `MPI_Send` (或 `MPI_Isend`) 所花费的时间由公式 $\alpha + \beta \times N$ 计算, 其中 α 和 β 为常数，N 为所发送的字节数，进一步假设接收消息花费的时间与发送消息一样，则每次迭代中子区域间通信所花费的时间为_____。**(7 分)**

二、 考虑下面的 C 程序段:

```
int myrank, nprocs, color, key;
MPI_Comm comm;
... ...
MPI_Init(&argc, &argv);
MPI_Comm_rank(MPI_COMM_WORLD, &myrank);
MPI_Comm_size(MPI_COMM_WORLD, &nprocs);
color = myrank / 5;
key = myrank;
MPI_Comm_split(MPI_COMM_WORLD, color, key, &comm);
... ...
```

假设程序运行时使用 9 个进程。

(1) 上述代码最后一个语句共创建了多少个不同的新通信器 comm? 各个新通信器包含了哪些进程? (按进程在新通信器中的顺序列出每个新通信器中的进程, 用 MPI_COMM_WORLD 中的进程号表示)。**(10 分)**

(2) 如果将“key = myrank”改成“key = (nprocs - myrank) / 2”, 对每个新通信器顺序列出它们所包含的进程。**(15 分)**

三、 考虑下面的 C 程序段:

```
int buffer[100];
MPI_Datatype type;

... ...
MPI_Type_vector(2, 10, 25, MPI_INT, &type);
MPI_Type_commit(&type);
... ...
```

(1) “MPI_Send(buffer, 1, type, ...)”发送哪些数据? **(10 分)**

(2) “MPI_Send(buffer, 2, type, ...)”发送哪些数据? **(15 分)**

提示: MPI_Type_vector 创建的数据类型的域 (extent) 是最后一个数据块的最后一个字节的位移加 1, 例如假设 sizeof(int) 为 4, 则 type 的域为 140。

四、 设通信器 comm 中的进程数为 np $= 2^k$ (k 为正整数)，当前进程在通信器 comm 中的进程号为 myrank。假设每个进程中有一个整数 (int) n，则下面的二叉树算法将各个进程中的 n 值加起来，结果在进程 0 中 (MPI_Reduce):

第 1 步 :

进程 0 与进程 1 相加，结果放在进程 0 中，
进程 2 与进程 3 相加，结果放在进程 2 中，
⋮
进程 np - 2 与进程 np - 1 相加，结果放在进程 np - 2 中。

第 2 步 :

进程 0 与进程 2 相加，结果放在进程 0 中，
进程 4 与进程 6 相加，结果放在进程 4 中，
⋮
进程 np - 4 与进程 np - 2 相加，结果放在进程 np - 4 中。

⋮

第 k 步 :

进程 0 与进程 np / 2 相加，结果放在进程 0 中。

(1) 给出上述算法的 C 语言代码。**(15 分)**

提示: 参考下面的代码

```
int i, step, m;
... ...
step = 1;
while (step < np) {
    if ((myrank % step) != 0) break /*continue*/;
    i = myrank / step;
    if ((i % 2) == 0) {
        接收 m;
        计算 n := n + m;
    }
    else {
        发送 n;
    }
    step *= 2;
}
```

(2) 将上述代码中的 `break` 改成 `continue` 程序是否正确？各进程的运行流程上有什么区别？ **(10 分)**

五、 (选答题) 改造上一题的代码使其适合于 `np` 不是 2^k 的情形。**(20 分)**