

中国科学技术大学

2015—2016 学年第一学期考试试卷

考试科目: 并行程序设计

得分: _____

学生所在系: _____ 姓名: _____ 学号: _____

一、 (1)分析以下循环中的依赖关系,并给出相应的迭代依赖图: (3×10=30 分)

```
for i = 2 to 10 do //循环 1
    for j = i to 10
        A[i,j] = ( A[i,j-1] + A[i-1,j] ) * 0.5;
    endfor
endfor
```

```
for i = 1 to 16 do // 循环 2
    A[i+3] = A[i] + B[i];
endfor
```

```
for k = 1 to 16 step 5 do // 循环 3 ,k 的循环步长为 5
    for i = k to min(16,i+4) do //设 min 为求最小值函数
        A[i+3] = A[i] + B[i]
    endfor
endfor
```

(2) 给出以上循环 2 的 OpenMP 并行实现 (10 分)

二、 给出两种 MPI 实现方案: 0 号进程采用行交叉划分方式, 将 $N \times N$ 的双精度数矩阵发送到 P 个进程 (设 N 能被 P 整除)。(20 分)

三、 给出 cannon 乘法在 4×4 进程拓扑下的 A、B 子块矩阵在初始对齐以及前三次移位后的分布示意图。(10 分)

四、 设复数类型定义如下:

```
typedef struct {
    double real, imag;
} Complex;
```

给出一个自定义归约函数及相关 MPI 代码, 完成复数类型的加法。(15 分)

五、 两多边形相交问题的一种朴素并行算法如下：

对于多边形 R 的每一条边,要确定其是否与多边形 Q 相交;如果 R 的边中有一条边与 Q 相交,那么就可以断定多边形 R 与 Q 是相交的。假设 R 有 n 条边, Q 有 m 条边, 总共有 p 个处理器 P_1, P_2, \dots, P_p 。对于 R 中的每条边依次判断是否与 Q 相交。

输入: 多边形 R 的 n 条边 E_1, E_2, \dots, E_n 的两个端点坐标集合 S_1 , 多边形 Q 的 m 条边 F_1, F_2, \dots, F_m 的两个端点坐标集合 S_2

输出: 两个多边形是否相交: true(两多边形相交); false(两多边形不相交)

Begin

(1) for $i=1$ to m do 将 F_i 广播到所有处理器上 end for

(2) for $j=1$ to n do 将 E_j 广播到所有处理器上 end for

(3) for all P_k where $1 \leq k \leq p$ par-do

for $i=1$ to $\left\lceil \frac{n}{p} \right\rceil$ do

for $j=1$ to m do

if intersects ($E_{i \times p + k}, F_j$) then result_k=true end if

end for

end for

end for

(4) 将各个处理器上 result_k 返回到主处理器, 如果其中有一个为真, 则两多边形相交, 否则两多边形不相交

End

给出上述算法的 MPI 实现 (15 分)

本试卷答题过程中可能用到的函数原型:

MPI_Comm_size(MPI_Comm comm, int *size)

MPI_Comm_rank(MPI_Comm comm, int *rank)

MPI_Reduce(void *sendbuf, void *recvbuf, int count, MPI_Datatype datatype, MPI_Op op, int root, MPI_Comm comm)

MPI_Bcast(void *buffer, int count, MPI_Datatype datatype, int root, MPI_Comm comm)

MPI_Gather(void *sendbuf, int sendcnt, MPI_Datatype sendtype, void *recvbuf, int recvcnt, MPI_Datatype recvtype, int root, MPI_Comm comm)

MPI_Send(void *buf, int count, MPI_Datatype datatype, int dest, int tag, MPI_Comm comm)

MPI_Recv(void *buf, int count, MPI_Datatype datatype, int source, int tag, MPI_Comm comm, MPI_Status *status)

MPI_Sendrecv(void *sendbuf, int sendcount, MPI_Datatype sendtype, int dest, int sendtag, void *recvbuf, int recvcnt, MPI_Datatype recvtype, int source, int recvtg, MPI_Comm comm, MPI_Status *status)

MPI_Type_vector(int count, int blocklength, int stride, MPI_Datatype old_type, MPI_Datatype *newtype_p)

MPI_Wait(MPI_Request *request, MPI_Status *status)

MPI_Type_contiguous(int count, MPI_Datatype oldtype, MPI_Datatype *newtype)

MPI_Op_create(MPI_User_function *function, int commute, MPI_Op *op)

MPI_Type_commit(MPI_Datatype *datatype)

typedef void MPI_User_function(void *invec, void *inoutvec, int *len, MPI_Datatype *datatype);