装订线 答题时不要超过此线

中 国 科 学 技 术 大 学

2009－2010 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、给出用基本的 **MPI_Send** 和 **MPI_Recv** 函数实现 **MPI_Allgather** 函数功能的伪代码。这三个函数的调用格式如下:（20 分）

（1） **MPI_Send(buf, count, datatype, dest, tag, comm)**

（2） **MPI_Recv(buf, count, datatype, source, tag, comm, status)**

（3） **MPI_Allgather**(sendbuf, sendcount, sendtype, recvbuf, recvcount, recvtype, comm)

二、以下是一个用 C 语言描述的计算 π 的串行程序。（30 分）

```
static long num_steps = 100000;
double step;
void main ()
{       int i; double x, pi, sum = 0.0;
        step = 1.0/(double) num_steps;
        for (i=1;i<= num_steps; i++){
           x = (i-0.5)*step;
           sum = sum + 4.0/(1.0+x*x);
        }
        pi = step * sum;
}
```

请给出**四种不同的 OpenMP 程序**实现来并行计算 π 。

三、以下程序是快速排序的一种 OpenMP 并行递归实现。请补充实线框内的程序代码。（20 分）

```
void qs(int *v, int first, int last) {
  int start[2], end[2], pivot, i, temp;
  // 以下为数组划分的代码
  if (first < last) {
     start[1] = first;
     end[0] = last;
     pivot = v[(first + last) / 2];
     while (start[1] <= end[0]) {
       while (v[start[1]] < pivot) start[1]++;
       while (pivot < v[end[0]])   end[0]--;
       if (start[1] <= end[0]) {
         temp = v[start[1]];
```

```
        v[start[1]] = v[end[0]];
        v[end[0]] = temp;
        start[1]++;
        end[0]--;
      }
    }
    start[0] = first;
    end[1]   = last;
// 以下实线框内为并行部分的代码，请补充！

  } // end-if
} // end-of-qs
```

四、请给出矩阵分块乘法的并行 **MPI** 实现代码（30 分）

矩阵乘法也可以用分块的思想实现并行，即分块矩阵乘法**(Block Matrix Multiplication)**，将矩阵 A（mXn）按行划分为 p 块(p 为处理器个数)，设 $u=\lceil m/p \rceil$，每块含有连续的 u 行向量，这些行块依次记为 A_0,A_1，…,A_{p-1}，分别存放在标号为 0,1,…，p-1 的处理器中。对矩阵 B（nXk）按列划分为 p 块，记 $v=\lceil k/p \rceil$，每块含有连续的 v 列向量，这些列块依次记为 B_0,B_1，…,B_{p-1}，分别存放在标号 0,1，…, p-1 为的处理器中。将结果矩阵 C 也相应地同时进行行、列划分，得到 $p\times p$ 个大小为 $u\times v$ 的子矩阵，记第 i 行第 j 列的子矩阵为 C_{ij}，显然有 $C_{ij}=A_i\times B_j$，其中，A_i 大小为 $u\times n$，B_j 大小为 $n\times v$。

开始，各处理器的存储内容如图**(a)**所示。此时各处理器并行计算 $C_{ii}=A_i\times B_i$ 其中 i=0,1,…,p-1，此后第 i 号处理器将其所存储的 B 的列块送至第 i-1 号处理器（第 0 号处理器将 B 的列块送至第 p-1 号处理器中，形成循环传送），各处理器中的存储内容如图**(b)**所示。它们再次并行计算 $C_{ij}=A_i\times B_j$，这里 $j=(i+1)\bmod p$。B 的列块在各处理器中以这样的方式循环传送 p-1 次并做 p 次子矩阵相乘运算，就生成了矩阵 C 的所有子矩阵。最终，编号为 i 的处理器的内部存储器存有 C 子矩阵 $C_{i0},C_{i1},\ldots,C_{i(p-1)}$。

处理器编号	存储内容	
0	A_0	B_0
1	A_1	B_1
2	A_2	B_2
•••	•••	•••
P-1	A_{p-1}	B_{p-1}

(a)

处理器编号	存储内容	
0	A_0	B_1
1	A_1	B_2
2	A_2	B_3
•••	•••	•••
P-1	A_{p-1}	B_0

(b)