

《并行计算》实验报告

# 天津大学

## MPI 实现矩阵转置算法



学    院 计算机科学与技术  
专    业 计算机科学与技术  
年    级 2012 级  
姓    名 王雨朦  
学    号 3012216083

2015 年 5 月 2 日

## 1. 实验内容

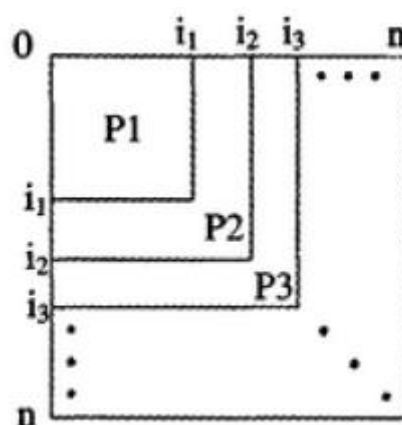
实用 MPI 并行编程技术计算对应的转置矩阵

## 2. 实验原理

### 1) 数学模型

直角划分方法。

如图：



转置分两步进行：第一步，将矩阵划分为大小相近的  $p$  个子块；第二步，对每一个子块进行转置。

### 2) 实现方法

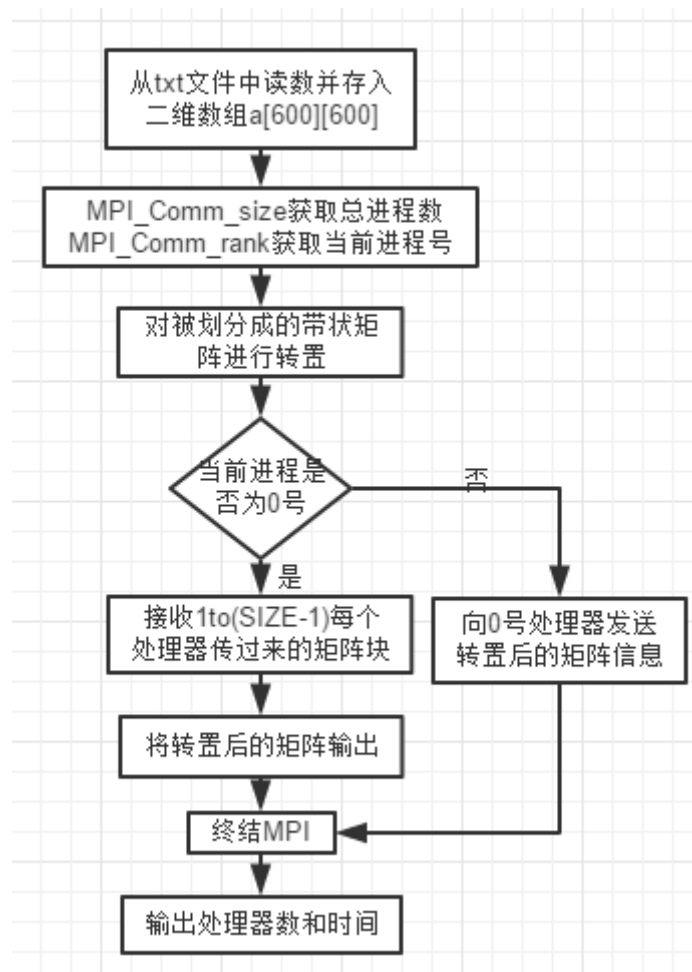
直角划分方法对矩阵的转置分两部分。第一步，所有子块内部转置；第二步，把第 1 到  $N-1$  个处理器子块传给 0 号处理器子块矩阵。

要实现 MPI 进行矩阵的转置，用直角划分方法比较容易。

进行矩阵块间的转置时，直接交换横纵坐标即可。再判断是否为 0 号处理器，如果是，就接收从 1 到  $N-1$  的转置后的块然后输出到“transmatrix.txt”文件。如果不是，就向 0 号矩阵发送经过转置的带状矩阵。

进行发送和接收时，将每个直角矩阵分为两部分进行。如果标记带状矩阵的左右边界分为  $L$ 、 $R$ ，则先发送  $[0, L]$  行矩阵，因为它的列范围是  $[L, R]$ 。再发送  $[L, R]$  行矩阵，它的列范围是  $[0, R]$

### 3. 程序流程



### 4. 实验结果及分析

#### 1) 实验结果数据

下图为文件 `matrix.txt` (600\*600) 的矩阵进行转置后的左上角一块 15\*15 矩阵的显示结果。跟上次实验结果相同。

```
[AK073@node128 ~/data]$ mpic++ -i_dynamic -o test test.cpp
[AK073@node128 ~/data]$ qsub test.pbs
25207.node126
[AK073@node128 ~/data]$ qstat -R

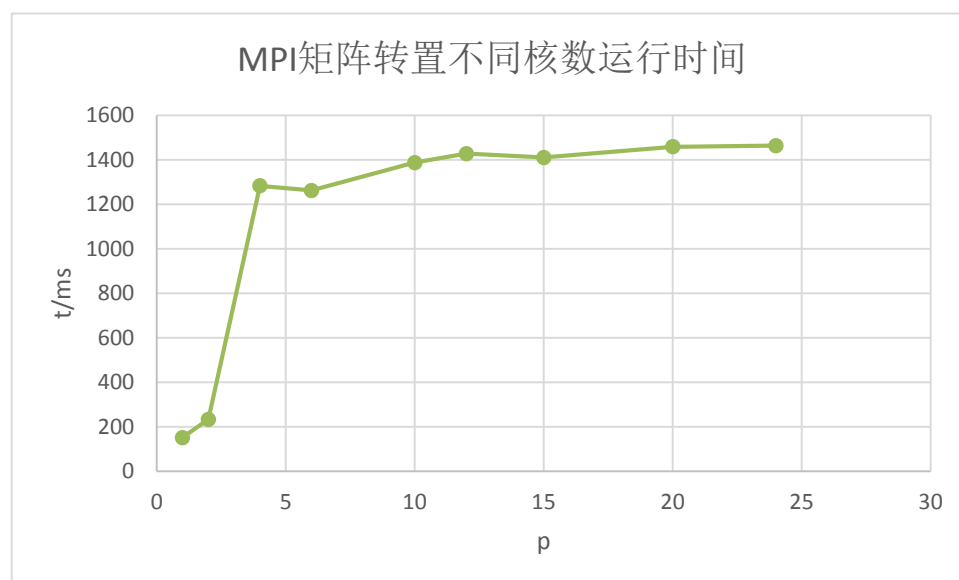
node126:

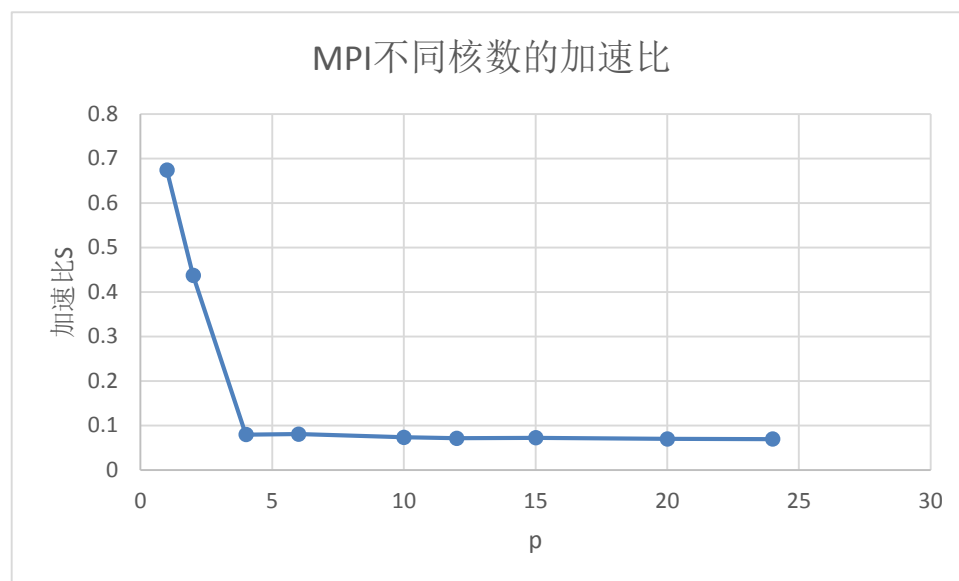
Job ID              Username      Queue      NDS    TSK      Req'd  Req'd   Elap
BIG  FAST   PFS                               Memory Time   S Time
-----
25207.node126      AK073        qstudent    2      4       --   02:00 C 00:00:01
--  --  --

[AK073@node128 ~/data]$ cat run.log
After transposed:
83 70 13 23 46 27 14 79 18 78 89 36 64 54 38
86 99 91 3 12 51 51 60 15 51 38 44 13 63 44
77 17 38 0 55 96 14 77 26 73 23 77 38 55 33
15 39 70 20 93 82 52 0 99 0 60 76 31 8 18
93 69 21 0 42 10 40 60 72 97 53 61 79 69 65
35 63 67 42 0 94 41 46 65 2 65 74 28 63 28
86 22 29 52 50 98 30 97 81 25 27 31 71 5 33
92 94 71 12 20 92 53 38 46 2 26 90 45 81 71
49 73 80 4 89 82 64 24 60 73 64 44 96 88 63
21 47 43 7 28 53 90 54 94 31 71 3 76 59 61
62 31 95 59 50 53 54 0 63 22 80 8 44 8 43
27 62 99 58 49 2 30 71 81 78 47 86 30 14 94
90 82 24 25 56 47 86 89 55 80 62 1 14 39 5
59 90 88 94 37 52 72 31 45 36 75 69 64 63 78
63 92 54 17 42 54 62 78 43 38 41 18 50 84 89
The core number is: 4
cost time: 221 ms
```

## 2) 加速比曲线

向集群申请不同的核数，得的数据绘制成曲线和加速比曲线图如下：





### 3) 实验结果分析:

由于数据量太小，用到 MPI 多核处理没有得到理想的实验结果。分配的核数越多得到的运行时间越长，猜想可能是虽然计算的时间减少了，但是在计算量很小的情况下，进程之间通信时的数据交换反而造成了更多的时间开销。

## 5. 实验总结

虽然没有得到理想的实验结果，但是第一次用 MPI+C++进行多核计算的实验，从一个小的实验开始对 MPI 六个基本命令进行了学习和运用。

这次由于用 MPI 进行多核处理，上次多线程进行矩阵转置的棋盘划分思想就显得略复杂，所以转而用了直角划分算法来做转置。相比之下，直角划分还是更容易理解，计算起来也相对简单。