

《并行计算》实验报告

天津大学

多线程实现矩阵转置算法



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 4 月 20 日

1. 实验内容

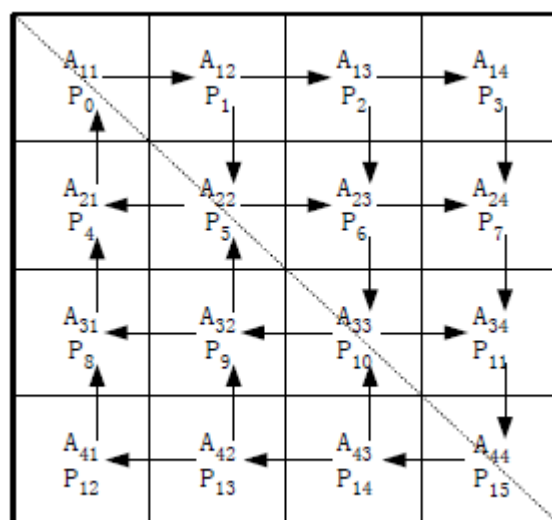
- 1) 实现矩阵转置算法(串行)
- 2) 多线程实现矩阵转置算法
- 3) 实现用不同的线程数 P 进行对已给矩阵(600*600)的转置。记录进行矩阵转置所用的时间，绘制加速比曲线并对结果进行分析

2. 实验原理

1) 数学模型

块棋盘划分方法：

假设处理器个数为 P ，编号为 $0, 1, \dots, P-1$ ，则将 N 阶矩阵 A 分成 P 个大小为 $m \times n$ 的子块， P 个子块组成一个 $\sqrt{P} \times \sqrt{P}$ 的子块矩阵。如图：



2) 实现方法

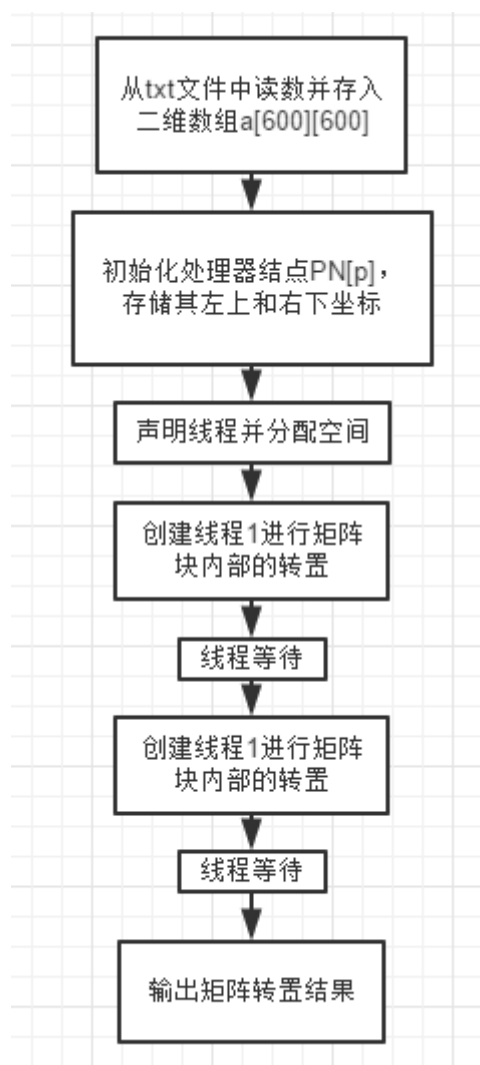
块棋盘划分方法对矩阵的转置分两部分。第一步，子块转置；第二步，处理器间局部转置。

要实现这一算法，先定义一个结构体，代表每个处理器结点，结构体内记录每个 $m \times m$ 矩阵的左上结点和右下结点的坐标。这样就可以简便又清晰的进行矩阵内部和矩阵块之间的转置。

进行矩阵内部转换时，思想同串行矩阵转置，巧妙利用矩阵块左右上下边界结点坐标，交换相应位置的数即可。

进行矩阵块间的转置时，先把处理器编号从一维转换到二维，在这个算法中当前编号为 N 的矩阵块对应的二维坐标为 $(N/\sqrt{P}, N\% \sqrt{P})$ 。从而交换横纵坐标，可以得到其应该交换的矩阵块对应的二维坐标为 $(N\% \sqrt{P}, N/\sqrt{P})$ ，编号为 $(N\% \sqrt{P}) * \sqrt{P} + N/\sqrt{P}$ 。再将这两个矩阵块内位于相同位置的数字进行交换即可。

3. 程序流程



4. 实验结果及分析

1) 实验结果数据

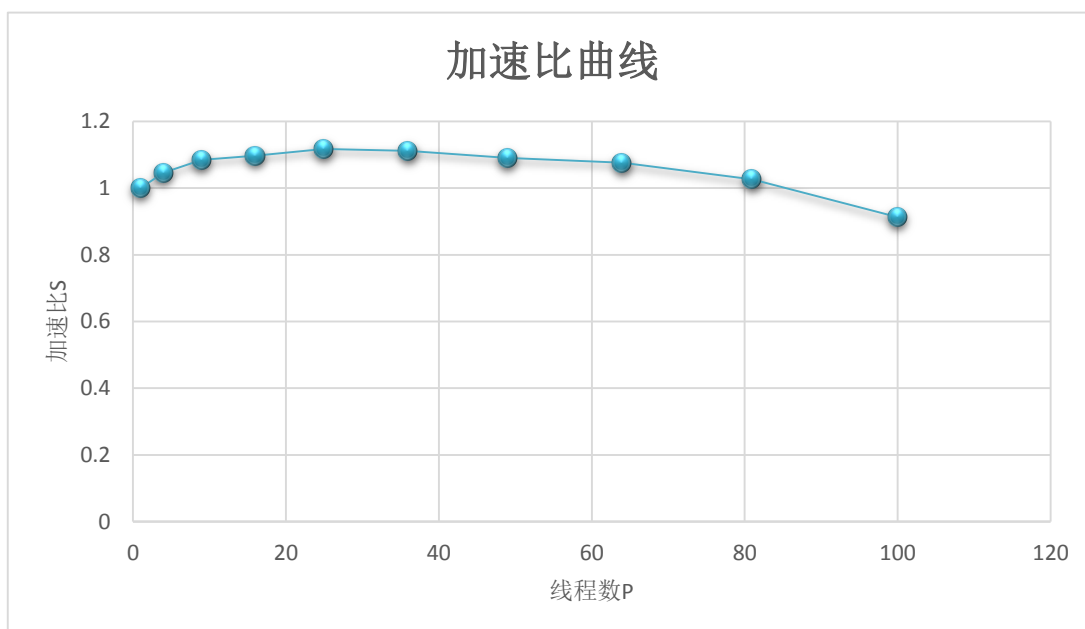
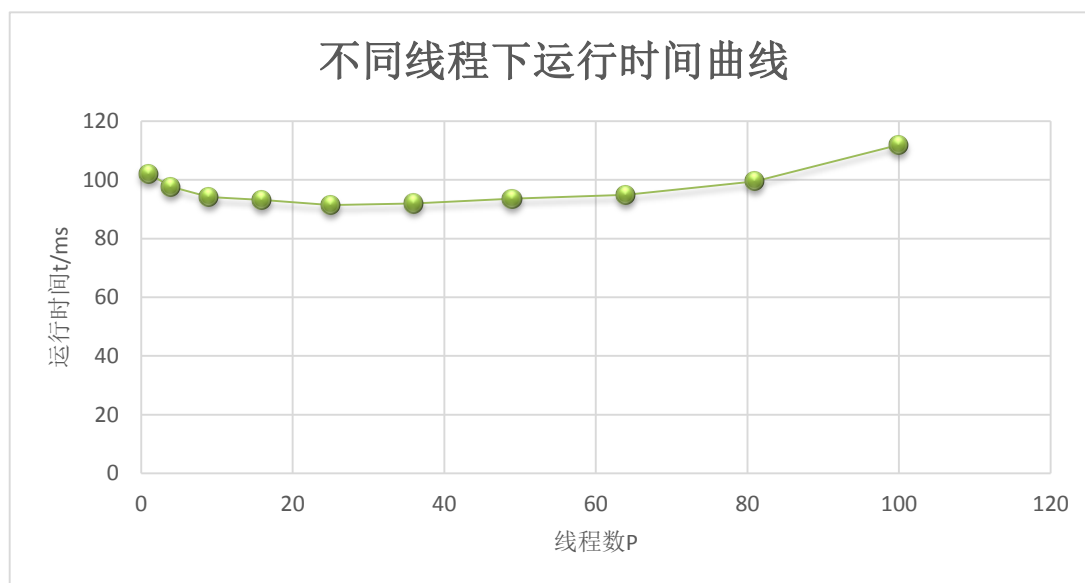
下图为文件 matrix.txt (600*600) 的矩阵进行转置前后的左上角一块 15*15 矩阵的显示结果

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

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
```

2) 加速比曲线

shell 脚本在集群上循环运行多线程进行矩阵转置的程序，并计算每种情况下进行矩阵转置的时间的平均值，得的数据绘制成离散加速比曲线图如下：



3) 实验结果分析:

进行矩阵转置的计算时间随着线程数 T 的增大, 先显著的减少, 然后缓慢的减少, 最后反而增大。可知, 分配的线程一味得多也并不是会起到不能很快加速程序计算的效果, 更糟糕的情况是有时候执行小程序时, 分配线程所用的时间加起来反而比串行执行程序本身所花费的时候更长。

5. 实验总结

通过本次实验，更加熟悉了在集群上进行的各种操作的指令。

通过一个简单的进行矩阵转置的并行程序，在各种情况下运行程序，记录并分析数据，绘制成曲线，加深了对并行计算的理解。并行计算在一定程度下可以加快程序的运行时间，但是如果并行的线程数过多，运行小程序时不仅是时间加快的效果不明显，有时候反而会比串行执行程序花费更多的时间。所以以后在自己运行实验程序时应该为程序分配合适的线程数，来起到加快运行时间又不浪费资源的目的。