

矩阵大小: 10x10
每个算法重复执行次数: 3000

-
1. 逐列访问算法: 7.9e-05 ms [正确]
 2. 缓存优化算法: 6.26667e-05 ms [正确]
-

加速比: 1.26064x

实验结论: 缓存优化的按行访问算法利用了空间局部性, 显著提高了内存访问效率

矩阵大小: 20x20
每个算法重复执行次数: 3000

-
1. 逐列访问算法: 0.000517667 ms [正确]
 2. 缓存优化算法: 0.000239667 ms [正确]
-

加速比: 2.15994x

实验结论: 缓存优化的按行访问算法利用了空间局部性, 显著提高了内存访问效率

矩阵大小: 30x30
每个算法重复执行次数: 3000

-
1. 逐列访问算法: 0.000621667 ms [正确]
 2. 缓存优化算法: 0.000268 ms [正确]
-

加速比: 2.31965x

实验结论: 缓存优化的按行访问算法利用了空间局部性, 显著提高了内存访问效率

矩阵大小: 40x40
每个算法重复执行次数: 3000

-
1. 逐列访问算法: 0.00136833 ms [正确]
 2. 缓存优化算法: 0.00046 ms [正确]
-

加速比: 2.97464x

实验结论: 缓存优化的按行访问算法利用了空间局部性, 显著提高了内存访问效率

wzsql@coffee:/mnt/d/Desktop/并行实验/lab2\$ g++ -o matrix_vector 001.cpp -O3
wzsql@coffee:/mnt/d/Desktop/并行实验/lab2\$./matrix_vector
矩阵大小: 40x40
每个算法重复执行次数: 3000

-
1. 逐列访问算法: 0.00229067 ms [正确]
 2. 缓存优化算法: 0.000803333 ms [正确]
-

加速比: 2.85145x

实验结论: 缓存优化的按行访问算法利用了空间局部性, 显著提高了内存访问效率

wzsql@coffee:/mnt/d/Desktop/并行实验/lab2\$ g++ -o matrix_vector 001.cpp -O3
wzsql@coffee:/mnt/d/Desktop/并行实验/lab2\$./matrix_vector
矩阵大小: 40x40
每个算法重复执行次数: 3000

-
1. 逐列访问算法: 0.00163267 ms [正确]
 2. 缓存优化算法: 0.000584667 ms [正确]
-

加速比: 2.79247x

实验结论: 缓存优化的按行访问算法利用了空间局部性, 显著提高了内存访问效率

矩阵大小：50x50

每个算法重复执行次数：3000

1. 逐列访问算法：0.00330333 ms [正确]

2. 缓存优化算法：0.00102667 ms [正确]

加速比：3.21753x

实验结论：缓存优化的按行访问算法利用了空间局部性，显著提高了内存访问效率

矩阵大小：60x60

每个算法重复执行次数：3000

1. 逐列访问算法：0.00241133 ms [正确]

2. 缓存优化算法：0.000778333 ms [正确]

加速比：3.09807x

实验结论：缓存优化的按行访问算法利用了空间局部性，显著提高了内存访问效率

wzsql@coffee: /mnt/d/Desktop/并行实验/lab2\$./matrix_vector

矩阵大小：70x70

每个算法重复执行次数：3000

1. 逐列访问算法：0.003438 ms [正确]

2. 缓存优化算法：0.00112933 ms [正确]

加速比：3.04427x

实验结论：缓存优化的按行访问算法利用了空间局部性，显著提高了内存访问效率

wzsql@coffee: /mnt/d/Desktop/并行实验/lab2\$./matrix_vector

矩阵大小：80x80

每个算法重复执行次数：3000

1. 逐列访问算法：0.00600933 ms [正确]

2. 缓存优化算法：0.002004 ms [正确]

加速比：2.99867x

实验结论：缓存优化的按行访问算法利用了空间局部性，显著提高了内存访问效率

wzsql@coffee: /mnt/d/Desktop/并行实验/lab2\$./matrix_vector

矩阵大小：90x90

每个算法重复执行次数：3000

1. 逐列访问算法：0.007986 ms [正确]

2. 缓存优化算法：0.002589 ms [正确]

加速比：3.08459x

实验结论：缓存优化的按行访问算法利用了空间局部性，显著提高了内存访问效率

矩阵大小：100x100

每个算法重复执行次数：3000

1. 逐列访问算法：0.008024 ms [正确]

2. 缓存优化算法：0.00283133 ms [正确]

加速比：2.834x

实验结论：缓存优化的按行访问算法利用了空间局部性，显著提高了内存访问效率

矩阵大小：200x200

每个算法重复执行次数：3000

1. 逐列访问算法：0.0373593 ms [正确]

2. 缓存优化算法：0.011403 ms [正确]

加速比：3.27627x

实验结论：缓存优化的按行访问算法利用了空间局部性，显著提高了内存访问效率

wzsql@coffee:/mnt/d/Desktop/并行实验/lab2\$ g++ -o matrix_vector 001.cpp -O3

wzsql@coffee:/mnt/d/Desktop/并行实验/lab2\$./matrix_vector

矩阵大小：200x200

每个算法重复执行次数：3000

1. 逐列访问算法：0.040517 ms [正确]

2. 缓存优化算法：0.0127303 ms [正确]

加速比：3.18271x

实验结论：缓存优化的按行访问算法利用了空间局部性，显著提高了内存访问效率

矩阵大小：200x200

每个算法重复执行次数：3000

1. 逐列访问算法：0.0321947 ms [正确]

2. 缓存优化算法：0.0109107 ms [正确]

加速比：2.95075x

实验结论：缓存优化的按行访问算法利用了空间局部性，显著提高了内存访问效率

wzsql@coffee:/mnt/d/Desktop/并行实验/lab2\$ g++ -o matrix_vector 001.cpp -O3

wzsql@coffee:/mnt/d/Desktop/并行实验/lab2\$./matrix_vector

矩阵大小：300x300

每个算法重复执行次数：3000

1. 逐列访问算法：0.094796 ms [正确]

2. 缓存优化算法：0.0271987 ms [正确]

加速比：3.48532x

实验结论：缓存优化的按行访问算法利用了空间局部性，显著提高了内存访问效率

wzsql@coffee:/mnt/d/Desktop/并行实验/lab2\$./matrix_vector

矩阵大小：400x400

每个算法重复执行次数：3000

1. 逐列访问算法：0.173972 ms [正确]

2. 缓存优化算法：0.0515957 ms [正确]

加速比：3.37183x

实验结论：缓存优化的按行访问算法利用了空间局部性，显著提高了内存访问效率

```
wzsql@coffee:/mnt/d/Desktop/并行实验/lab2$ ./matrix_vector
```

矩阵大小：400x400

每个算法重复执行次数：3000

1. 逐列访问算法：0.160505 ms [正确]

2. 缓存优化算法：0.0625693 ms [正确]

加速比：2.56523x

实验结论：缓存优化的按行访问算法利用了空间局部性，显著提高了内存访问效率

```
wzsql@coffee:/mnt/d/Desktop/并行实验/lab2$ g++ -o matrix_vector 001.cpp -O3
```

```
wzsql@coffee:/mnt/d/Desktop/并行实验/lab2$ ./matrix_vector
```

矩阵大小：400x400

每个算法重复执行次数：3000

1. 逐列访问算法：0.186748 ms [正确]

2. 缓存优化算法：0.0497703 ms [正确]

加速比：3.75219x

实验结论：缓存优化的按行访问算法利用了空间局部性，显著提高了内存访问效率

```
wzsql@coffee:/mnt/d/Desktop/并行实验/lab2$ ./matrix_vector
```

矩阵大小：500x500

每个算法重复执行次数：3000

1. 逐列访问算法：0.28937 ms [正确]

2. 缓存优化算法：0.084454 ms [正确]

加速比：3.42637x

实验结论：缓存优化的按行访问算法利用了空间局部性，显著提高了内存访问效率

```
wzsql@coffee:/mnt/d/Desktop/并行实验/lab2$ g++ -o matrix_vector 001.cpp
```

```
wzsql@coffee:/mnt/d/Desktop/并行实验/lab2$ g++ -o matrix_vector 001.cpp
```

```
wzsql@coffee:/mnt/d/Desktop/并行实验/lab2$ ./matrix_vector
```

矩阵大小：500x500

每个算法重复执行次数：3000

1. 逐列访问算法：0.289668 ms [正确]

2. 缓存优化算法：0.072193 ms [正确]

加速比：4.01241x

实验结论：缓存优化的按行访问算法利用了空间局部性，显著提高了内存访问效率

```
wzsql@coffee:/mnt/d/Desktop/并行实验/lab2$ ./matrix_vector
```

矩阵大小：500x500

每个算法重复执行次数：3000

1. 逐列访问算法：0.320127 ms [正确]

2. 缓存优化算法：0.0786477 ms [正确]

加速比：4.0704x

实验结论：缓存优化的按行访问算法利用了空间局部性，显著提高了内存访问效率

```
wzsql@coffee:/mnt/d/Desktop/并行实验/lab2$ g++ -o matrix_vector 001.cpp -O3
```

```
wzsql@coffee:/mnt/d/Desktop/并行实验/lab2$ ./matrix_vector
```

矩阵大小：500x500

每个算法重复执行次数：3000

1. 逐列访问算法：0.341071 ms [正确]

2. 缓存优化算法：0.0840187 ms [正确]

加速比：4.05947x

实验结论：缓存优化的按行访问算法利用了空间局部性，显著提高了内存访问效率

```
wzsql@coffee:/mnt/d/Desktop/并行实验/lab2$ ./matrix_vector
```

矩阵大小：600x600

每个算法重复执行次数：3000

1. 逐列访问算法：0.552139 ms [正确]

2. 缓存优化算法：0.122344 ms [正确]

加速比：4.51299x

实验结论：缓存优化的按行访问算法利用了空间局部性，显著提高了内存访问效率

```
wzsql@coffee:/mnt/d/Desktop/并行实验/lab2$
```

```
wzsql@coffee:/mnt/d/Desktop/并行实验/lab2$ g++ -o matrix_vector 001.cpp -O3
```

矩阵大小：600x600

每个算法重复执行次数：3000

1. 逐列访问算法：0.467211 ms [正确]

2. 缓存优化算法：0.133834 ms [正确]

加速比：3.49097x

实验结论：缓存优化的按行访问算法利用了空间局部性，显著提高了内存访问效率

```
wzsql@coffee:/mnt/d/Desktop/并行实验/lab2$ g++ -o matrix_vector 001.cpp -O3
```

```
wzsql@coffee:/mnt/d/Desktop/并行实验/lab2$ ./matrix_vector
```

矩阵大小：600x600

每个算法重复执行次数：3000

1. 逐列访问算法：0.521331 ms [正确]

2. 缓存优化算法：0.140919 ms [正确]

加速比：3.69951x

```
wzsql@coffee:/mnt/d/Desktop/并行实验/lab2$ ./matrix_vector
```

矩阵大小：700x700

每个算法重复执行次数：3000

1. 逐列访问算法：0.764189 ms [正确]

2. 缓存优化算法：0.210592 ms [正确]

加速比：3.62877x

实验结论：缓存优化的按行访问算法利用了空间局部性，显著提高了内存访问效率

```
wzsql@coffee:/mnt/d/Desktop/并行实验/lab2$ g++ -o matrix_vector 001.cpp -O3
```

```
wzsql@coffee:/mnt/d/Desktop/并行实验/lab2$ ./matrix_vector
```

矩阵大小：700x700

每个算法重复执行次数：3000

1. 逐列访问算法：0.763936 ms [正确]

2. 缓存优化算法：0.189533 ms [正确]

加速比：4.03062x

实验结论：缓存优化的按行访问算法利用了空间局部性，显著提高了内存访问效率

```
wzsql@coffee:/mnt/d/Desktop/并行实验/lab2$ ./matrix_vector
```

矩阵大小：700x700

每个算法重复执行次数：3000

1. 逐列访问算法：0.839621 ms [正确]

2. 缓存优化算法：0.176231 ms [正确]

加速比：4.76432x

实验结论：缓存优化的按行访问算法利用了空间局部性，显著提高了内存访问效率


```
wzsql@coffee:/mnt/d/Desktop/并行实验/lab2$ ./matrix_vector
```

矩阵大小：800x800

每个算法重复执行次数：3000

1. 逐列访问算法：1.04164 ms [正确]

2. 缓存优化算法：0.249769 ms [正确]

加速比：4.17042x

```
wzsql@coffee:/mnt/d/Desktop/并行实验/lab2$ ./matrix_vector
```

矩阵大小：800x800

每个算法重复执行次数：3000

1. 逐列访问算法：1.06765 ms [正确]

2. 缓存优化算法：0.219287 ms [正确]

加速比：4.86874x

实验结论：缓存优化的按行访问算法利用了空间局部性，显著提高了内存访问效率

矩阵大小：900x900

每个算法重复执行次数：3000

1. 逐列访问算法：1.40015 ms [正确]

2. 缓存优化算法：0.334992 ms [正确]

加速比：4.17966x

实验结论：缓存优化的按行访问算法利用了空间局部性，显著提高了内存访问效率

```
wzsql@coffee:/mnt/d/Desktop/并行实验/lab2$ g++ -o matrix_vector 001.cpp -O3
```

```
wzsql@coffee:/mnt/d/Desktop/并行实验/lab2$ ./matrix_vector
```

矩阵大小：900x900

每个算法重复执行次数：3000

1. 逐列访问算法：1.42185 ms [正确]

2. 缓存优化算法：0.300705 ms [正确]

加速比：4.7284x

实验结论：缓存优化的按行访问算法利用了空间局部性，显著提高了内存访问效率

```
wzsql@coffee:/mnt/d/Desktop/并行实验/lab2$ ./matrix_vector
```

矩阵大小：900x900

每个算法重复执行次数：3000

1. 逐列访问算法：1.47961 ms [正确]

2. 缓存优化算法：0.341011 ms [正确]

加速比：4.33891x

实验结论：缓存优化的按行访问算法利用了空间局部性，显著提高了内存访问效率

```
wzsql@coffee:/mnt/d/Desktop/并行实验/lab2$ ./matrix_vector
```

矩阵大小：1000x1000

每个算法重复执行次数：3000

1. 逐列访问算法：1.9609 ms [正确]

2. 缓存优化算法：0.404769 ms [正确]

加速比：4.84449x

实验结论：缓存优化的按行访问算法利用了空间局部性，显著提高了内存访问效率

矩阵大小：1000x1000
每个算法重复执行次数：3000

-
1. 逐列访问算法：1.63954 ms [正确]
 2. 缓存优化算法：0.429511 ms [正确]
-

加速比：3.81723x

实验结论：缓存优化的按行访问算法利用了空间局部性，显著提高了内存访问效率

```
wzsql@coffee:/mnt/d/Desktop/并行实验/lab2$ g++ -o matrix_vector 001.cpp -O3
```

```
wzsql@coffee:/mnt/d/Desktop/并行实验/lab2$ ./matrix_vector
```

矩阵大小：1000x1000
每个算法重复执行次数：3000

-
1. 逐列访问算法：2.08858 ms [正确]
 2. 缓存优化算法：0.46926 ms [正确]
-

加速比：4.45081x

实验结论：缓存优化的按行访问算法利用了空间局部性，显著提高了内存访问效率