**ECE5720 Homework 2**

Jianhua Fan <jf773>

**Experiment**

1. For column sort

Determine the optimal number of thread (using 4 cores) (m = 256, n = 256)

|  |  |  |  |
| --- | --- | --- | --- |
| Number of thread | 2 | 4 | 8 |
| Time (ms) | 4113 | 3720 | 3582 |

The optimal number of threads are 8. So will use 8 threads for following comparison.

When m = 256, number of thread = 8

|  |  |  |  |
| --- | --- | --- | --- |
| n | 256 | 512 | 1024 |
| Time (ms) | 3769 | 4035 | 4499 |

When n = 256, number of thread = 8

|  |  |  |  |
| --- | --- | --- | --- |
| m | 256 | 512 | 1024 |
| Time (ms) | 3769 | 4285 | 4589 |

When m and n both increase, number of thread = 8

|  |  |  |  |
| --- | --- | --- | --- |
| m and n | 256 | 512 | 1024 |
| Time (ms) | 3769 | 8227 | 27018 |

1. For block sort

Determine the optimal number of thread (using 4 cores) (m = 256, n = 256)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Number of thread | 2 | 4 | 8 | 16 |
| Time (ms) | 4207 | 3982 | 3462 | 3963 |

The optimal number of threads are 8. After number of thread exceeds 8, the total time increases, so will use 8 threads for following comparison.

When m = 256, number of thread = 8

|  |  |  |  |
| --- | --- | --- | --- |
| n | 256 | 512 | 1024 |
| Time (ms) | 3462 | 3749 | 4547 |

When n = 256, number of thread = 8

|  |  |  |  |
| --- | --- | --- | --- |
| m | 256 | 512 | 1024 |
| Time (ms) | 3462 | 3656 | 4317 |

When m and n both increase, number of thread = 8

|  |  |  |  |
| --- | --- | --- | --- |
| m and n | 256 | 512 | 1024 |
| Time (ms) | 3462 | 8227 | 20477 |

**Observation**

From above tables, we can see that when m or n increases, the execution time of column sort or block sort will increase a little bit, but compared to the increased size of matrix, the multiple threads of openmp are actually speeding up the matrix operations.