Mean:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Threads | 1 | 2 | 4 | 8 | 16 | 32 | 64 | 128 | 512 | 1024 | 2048 |
| Time (ms) | 19 | 20 | 26 | 37 | 29 | 32 | 37 | 58 | 106 | 202 | 219 |

Median:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Threads | 1 | 2 | 4 | 8 | 16 | 32 | 64 | 128 | 512 | 1024 | 2048 |
| Time(ms) | 92395 | 44103 | 21946 | 10407 | 4832 | 2529 | 1601 | 1364 | 4455 | 8970 | 14175 |

Merge sort complexity is **O(n log n),** which will run faster as n gets smaller. Increasing the number of threads (**k**) reduces n per thread, which in turn reduces the running time for the sorting and merging. However, as **k** > 128, the cost of context switching is higher than the benefit of a shorter sorting time, in addition to additional number of iterations the program must run through to merge **k** arrays. This results in a slower running time as **k** > 128.