Sys specs :

Hp
Processors: 8th Generation Intel Core i5, Quad Core, 2.4 GHz Clock Speed
8 GB DDR4 RAM
1TB RAM SSD

Threading :

file is of 1000000 bytes
time :(10 threads)
1. 2M = 6142msec
2. 1M= 3687.42 msec
3. 3M = 7091 msec
time:(8 threads)
1. 3M : 5982msec
2. 2M : 5869 msec
3. 1M : 3599 msec
file is of 500000 bytes
time :(10 threads)
1. 3M : 6403 msec
file is of 200000000 bytes
1. 3M: :801865 msec

without threading :

file is of 12345600 bytes
3M : 58400.4 msec
2M : 29266.2 msec
1M :15656.8 msec
file is of 500000000 bytes
3M: 545524 msec
2M : 502760 msec
file is of 500000000 bytes
3M :331210msec
file is of 200000000 bytes
3M : 191814 msec

observations :
when i increased the main memory the time starts gradually increasing it is because may be the number of disk operations decreases but here the disk reading bites are remained constant , so here the array sizes which are 3-d(i declared 3d) because of their higher sizes they are making the programe slower than expected ,

and if the main memory is 100MB the programes becomes sooo slow in the phase2 so that even after 30mins the execution will be going on but termination didnot happen

In threading the phase2 is expected to be much slower but here it wont go that much slow because of the decreased array sizes , it becomes even faster in phase2 and in phase1 it is expected to become faster but the calling threads causes a little barrier making the threading not much useful, but the write and sorting is done in threading , which made it some what faster but not as expected

commands :
files :
start.cpp -->
g++ --std=c++14 strart.cpp -o sort1
./sort1 input.txt output.txt 2 asc a
lil.cpp -->

g++ --std=c++17 lil.cpp -lpthread -o sort2
./sort2 input.txt output1.txt 10 2 asc a