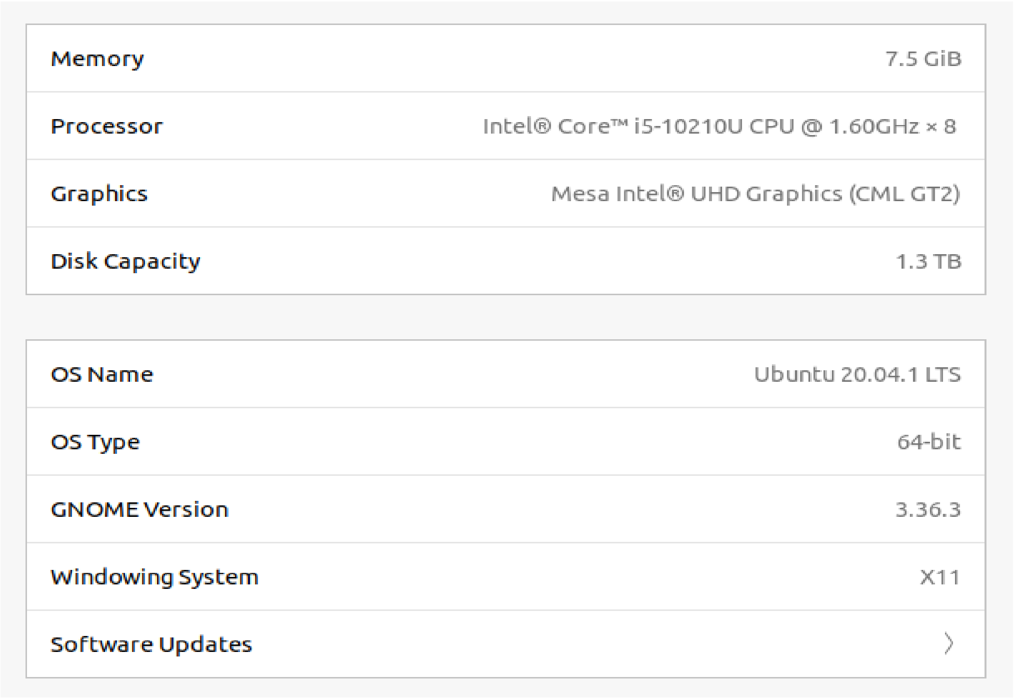
Two Phase Merge Sort Analysis

System Configuration:



Memory	7.5 GiB
Processor	Intel® Core™ i5-10210U CPU @ 1.60GHz × 8
Graphics	Mesa Intel® UHD Graphics (CML GT2)
Disk Capacity	1.3 TB

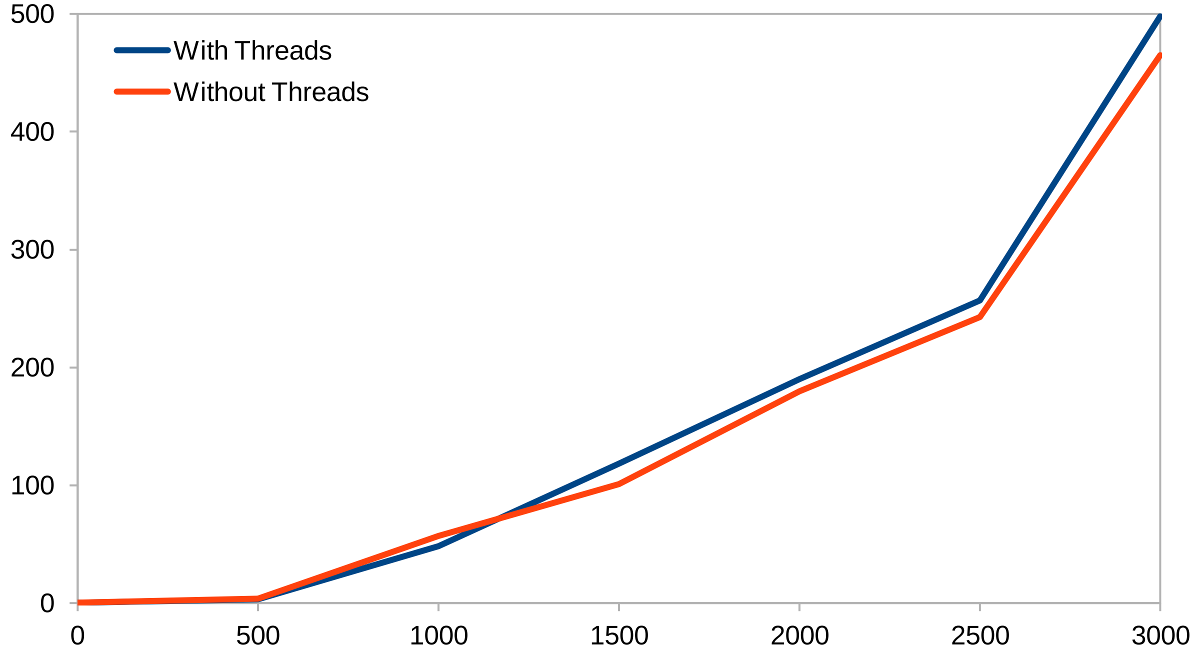
OS Name	Ubuntu 20.04.1 LTS
OS Type	64-bit
GNOME Version	3.36.3
Windowing System	X11
Software Updates	〉

Observations:

Observation #1: Varying file size with constant memory

Memory limit	Input File Size	Execution time with threads (seconds)	Execution time without threads (seconds)
100 MB	5MB	0.30	0.37
100 MB	50MB	3.41	3.64
100 MB	500MB	48.09	56.91
100 MB	1GB	118.41	101.20
100 MB	2GB	257	240
100 MB	3GB	498	465

- Sorting order: Ascending
- Columns used for sorting (in order): c1 c3
- No. of threads used: 5



Inference

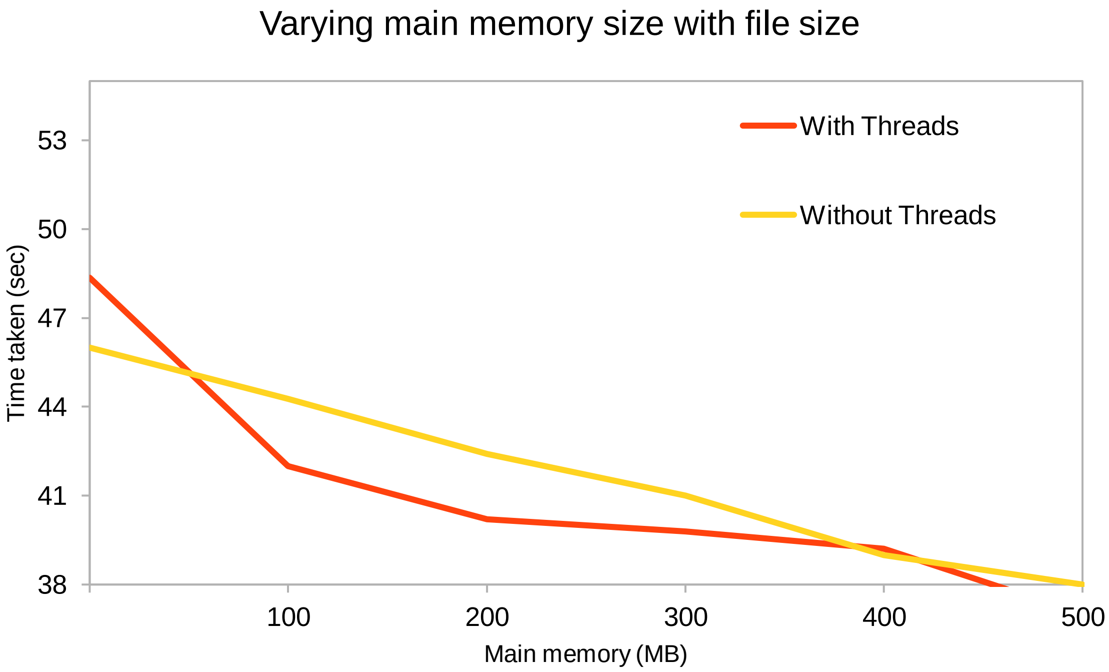
From above graph we infer that:

- Execution time with threads is less than that without threads upto a certain input file size, after which the program without threads performs better(in terms of execution time).
- Reason - as input file size increases and the amount of available main memory is constant, the number of intermediate files increases. This results in overhead, more jobs per cpu core. Hence, more time is spent on context switches and file IO.

Observation 2: Varying memory with constant file size

Main memory size	Execution time with threads (seconds)	Execution time without threads (seconds)
25MB	48.36	46.03
100MB	42.01	44.26
250MB	40.20	41.41
500MB	39.40	39.01
700MB	37.20	38.04

- File size: 500MB
- Sorting order: Ascending
- Columns used for sorting (in order): c1 c3
- No. of threads used: 5



Inference

From above graph we infer that:

- As the amount of main memory increases, the execution time in both cases (with and without threading) decreases.
- Also, the program with threads gives less execution time than that without threads.