Data Systems

Assignment-2

- **System Configuration**

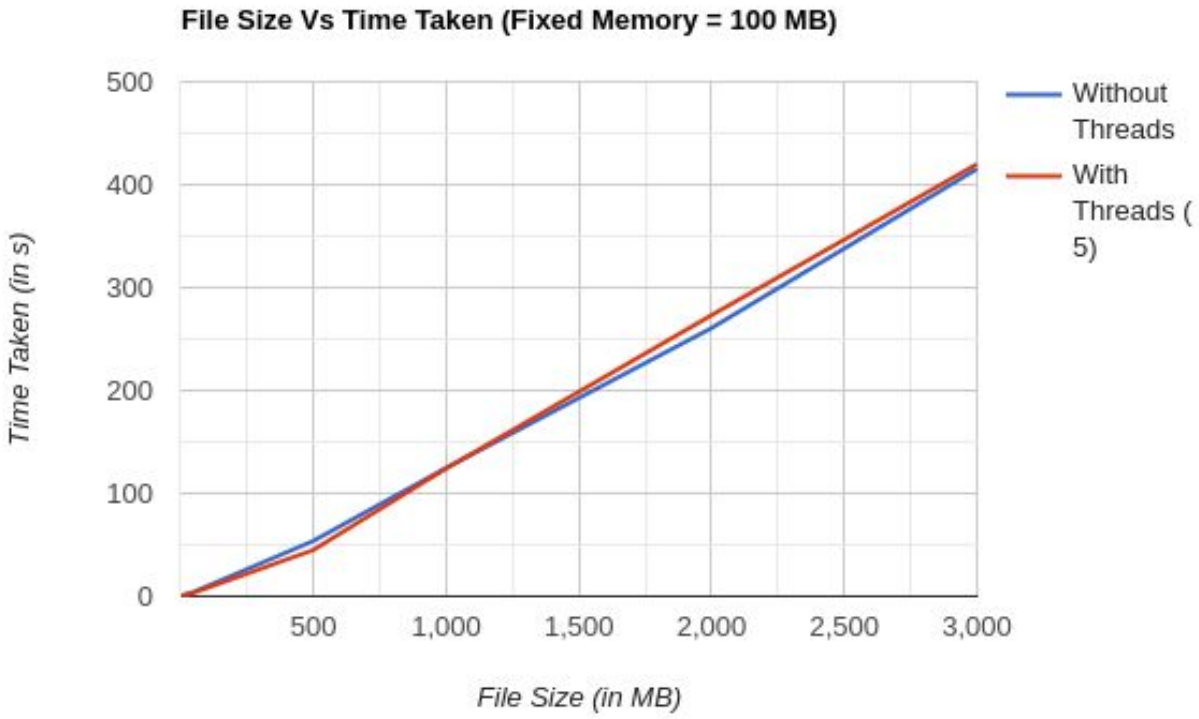
  Processor : Intel® Core™ i7-6500U CPU @ 2.50GHz × 4
  Memory : 15.5 GiB
  Disk Capacity : 1.0 TB
  Base System : Ubuntu 20.04 LTS 64-bit

- **Observations**

1) **Varying FileSize with constant memory :**

   Main Memory Size = 100MB

File Size (In MB)	Time Taken (In s) (Without Threads)	Time Taken (In s) (With Threads count=5)
5	0.39	0.42
50	4.47	3.68
500	54.01	44.93
1000	124.63	124.36
2000	260.73	273.27
3000	415.23	420.19



Explanation :

a) As the file size increases with main memory size kept as constant, the number of partitions **increases** resulting in longer processing times in phase1 and phase2 operations involving creating partitions, sorting them and writing the sorted records into a new file.

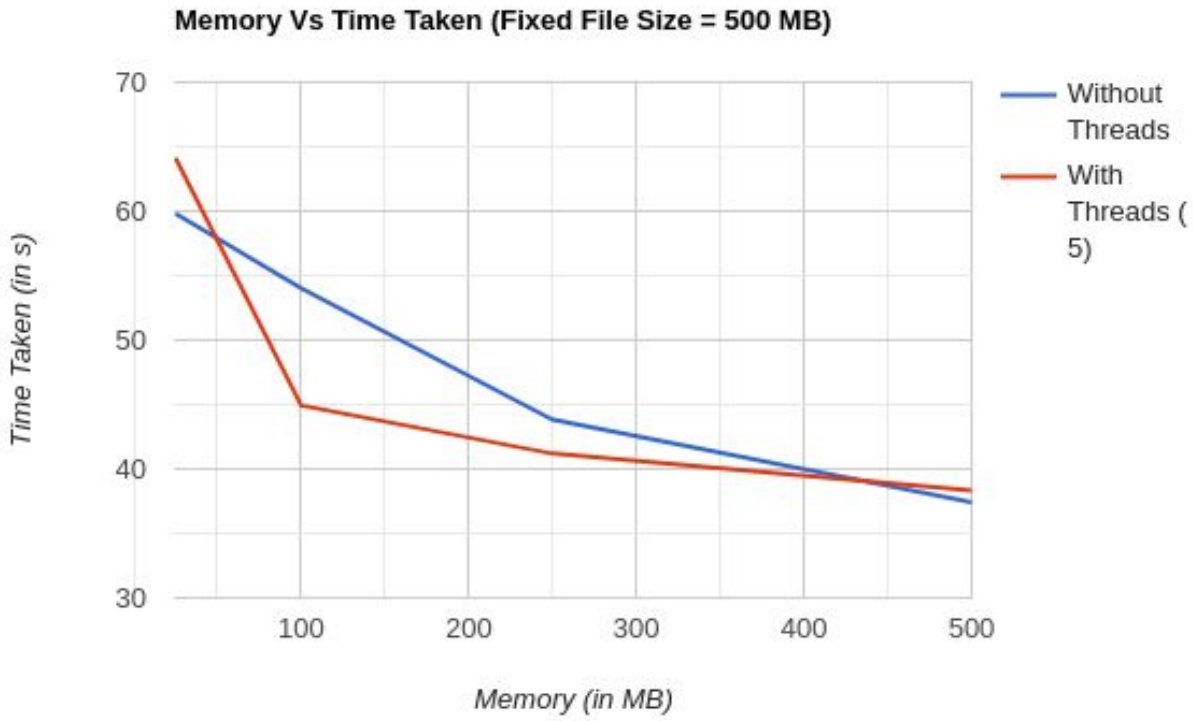
b) As per my observation, the introduction of threads led to marginal better performance while operating on file sizes upto 1000 MB since the load of generating different partitions get distributed among different threads (5 in this case) while for file sizes greater than 1500 MB, the overhead involved in shared resource consumption (that is reading from the same input file which is significantly bigger in size) is **high** which makes the distribution of workload among different threads less significant. More or less, the performance of the program

with and without threads are almost comparable as seen for file sizes upto 3000 MB while keeping main memory constant.

2) **Varying memory with constant FileSize :**

File Size = 500 MB

Memory size (In MB)	Time Taken (In s) (Without Threads)	Time Taken (In s) (With Threads count=5)
25	59.80	64.13
100	54.01	44.93
250	43.83	41.19
500	37.41	38.33



Explanation :

a) As per my observation, the overhead involved in shared resource consumption (that is reading from the same input file which is significantly bigger in size) remains constant for varying main memory sized partitions and since the file size is kept as constant and with increase in main memory sized partitions, the number of partitions decrease resulting in shorter processing times for the implementations involving with and without threads.