

REPORT

Sort on Single Shared Memory Node

PROGRAMMING ASSIGNMENT 02A CS 553 CLOUD COMPUTING

PROBLEM STATEMENT:

- The aim for this assignment was to sort the 2GB and 20GB files that was generated using gensort. Also the experiment was required to be done using multiple threads. Verification of the results obtained from the above experiments to be done using valsort.
- The assignment also required to sort the 2GB and 20GB files by using the linux sort. The output which is the sorted file of 2GB and 20GB is also to be validated or double checked whether it is sorted or not using valsort.
- All the above experiments were performed single shared memory node
- Sorting should be by external sort

APPROACH USED FOR SHARED MEMORY:

- I have written the code in java
- The following shows the approach I have used to sort 2 GB and 20 GB file using external sort
- I have written fileOpening() function which is used to open the file that is to be sorted i.e. 2GB or 20 GB file
- Thereafter the input file is split into chunks depending on the available memory
- And then each chunk is sorted for which I have created different function
- Then each of the sorted files are merged using a function mergeSortedFiles()
- The above function fetches the first n records depending upon the memory available from each of the chunks into memory
- The smallest value is determined iteratively among the records fetched in the above step and is written to an output file

- The above steps are repeated until one of the input buffers run out of records, at which point, a fresh set of n records obtained from the file corresponding to the buffer
- The above steps are executed iteratively until we obtain a sorted output file.

LINUX SORT:

- The sorting of 2GB and 20GB of files was done using linux sort also
- The sorted 2GB and 20GB files were then validated using the valsort
- The command I used to sort the 2 GB and 20 GB file was :

```
LC_ALL=C sort /input/data-20GB.in 1>/tmp/20GBsortLinux.log
```

- And to validate the following was used:

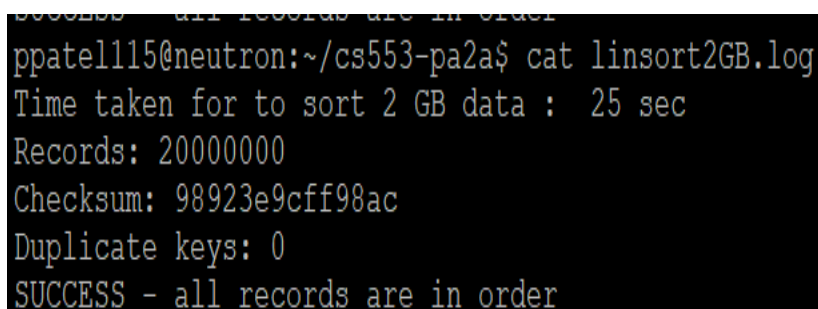
```
valsort /tmp/20GBsortLinux.log
```

- The linsort2GB.slurm and linsort20GB.slurm shows how I have executed linux sort and validated it

LINUX SORT PERFORMANCE:

- Following screenshots shows the time taken for linux sorting and whether the output was sorted or not:

1. Linux sort for 2 GB and valsort on 2GB:



```
ppatell115@neutron:~/cs553-pa2a$ cat linsort2GB.log
Time taken for to sort 2 GB data : 25 sec
Records: 20000000
Checksum: 98923e9cff98ac
Duplicate keys: 0
SUCCESS - all records are in order
```

Fig 1 : log file for linux sort and valsort – 2GB

- Fig 1 is the log file that I created using mysort code that has the time taken for linux sort on 2GB file which is 25 sec and the output of the valsort which shows that all the records are in order and also the duplicate key is 0

2. Linux sort for 20 GB and valsort on 20 GB:

```
ppatel115@neutron:~/cs553-pa2a$ cat linsort20GB.log
Time taken for to sort 20 GB data : 402 sec
Records: 200000000
Checksum: 5f5cc94518a4203
Duplicate keys: 0
SUCCESS - all records are in order
ppatel115@neutron:~/cs553-pa2a$
```

Fig 2 : log file for linux sort and valsort – 20 GB

- Fig 2 is the log file that I created using mysort code that has the time taken for linux sort on 20 GB file which is 402 sec and the output of the valsort which shows that all the records are in order and also the duplicate key is 0

PERFORMANCE EVALUATION OF TERASORT TABLE:

Performance evaluation of TeraSort				
Experiment	Shared Memory (1VM 2GB)	Linux Sort (1VM 2GB)	Shared Memory (1VM 20GB)	Linux Sort (1VM 20GB)
Compute Time (sec)	73.629	25	1013.88	402
Data Read (GB)	4	4	40	40
Data Write (GB)	4	4	40	40
I/O Throughput (MB/sec)	108.6528406	320	78.90480136	199.0049751

Table 1 : Performance evaluation of terasort

- Table 1 represent all the values that I obtained for mysort and linsort which included compute time, Throughput, Read and write
- 4 GB is being read and also 4 GB is written in mysort as I have 2 reads and 2 writes for 2GB
- 40 GB is being read and also 40 GB is written in mysort as I have 2 reads and 2 writes for 20 GB too
- Also linux sort has 2 reads hence 4 GB read and also 4GB written for 2GB file
- Also linux sort has 2 writes hence 40 GB read and also 40 GB written for 20GB file

Mysort EXPERIMENT RESULTS:

- Following are the results that I obtained and the comparison with the linux sort:

```
papatell115@neutron:~/cs553-pa2a$ cat mysort2GB.log
Execution/Compute time for 2 GB file sort = 73.629 sec
Records: 20000000
Checksum: 9894749b4557d4
Duplicate keys: 0
SUCCESS - all records are in order
papatell115@neutron:~/cs553-pa2a$ █
```

Fig 3: log file for mysort and valsort - 2GB

- Fig 3 shows the log file which I generate in mysort code which has the time 2GB file takes to get sorted and also the output of the valsort on my 2GB sorted file. All the records were sorted with 0 Duplicate key with compute time of 73.629 sec

```
papatell115@neutron:~/cs553-pa2a$ cat mysort20GB.log
Execution/Compute time for 20 GB file sort = 1013.88 sec
Records: 200000000
Checksum: 5f5d195b71ce0bd
Duplicate keys: 0
SUCCESS - all records are in order
papatell115@neutron:~/cs553-pa2a$ █
```

Fig 4: log file for mysort and valsort - 20GB

- Fig 4 shows the log file which I generate in mysort code which has the time 20GB file takes to get sorted and also the output of the valsort on my 20GB sorted file. Here too all the records were sorted with 0 Duplicate key with compute time of 1013.88 sec

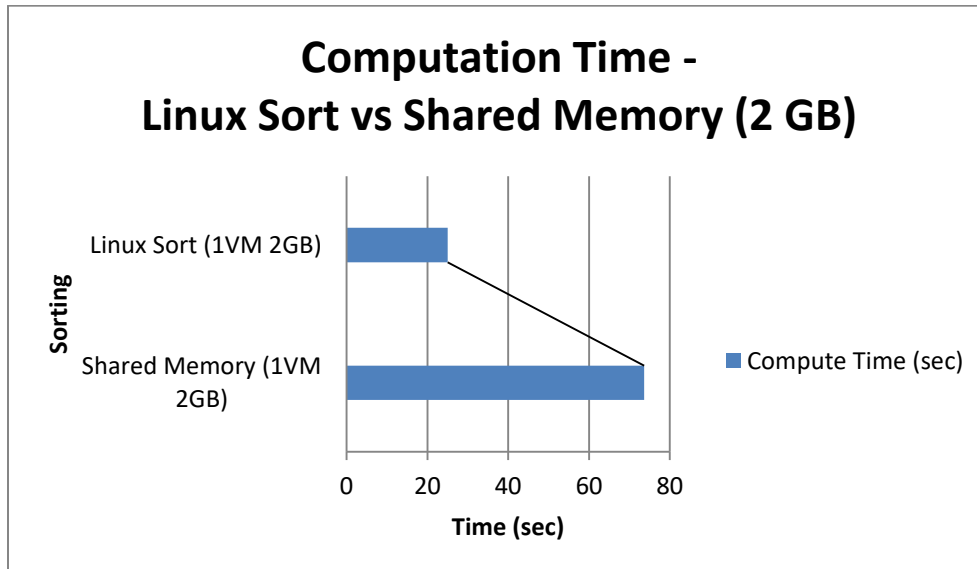


Fig 5 : Computation time - Linux Sort vs shared Memory - 2GB

- Fig 5 compares the compute time for linux sort and shared memory for 2GB file. The time taken for linux sort on 2 GB file was almost 48 seconds less than the sorting I performed as linux sort uses efficient algorithm to sort the records by using more threads and also depends in the language used and many other factors

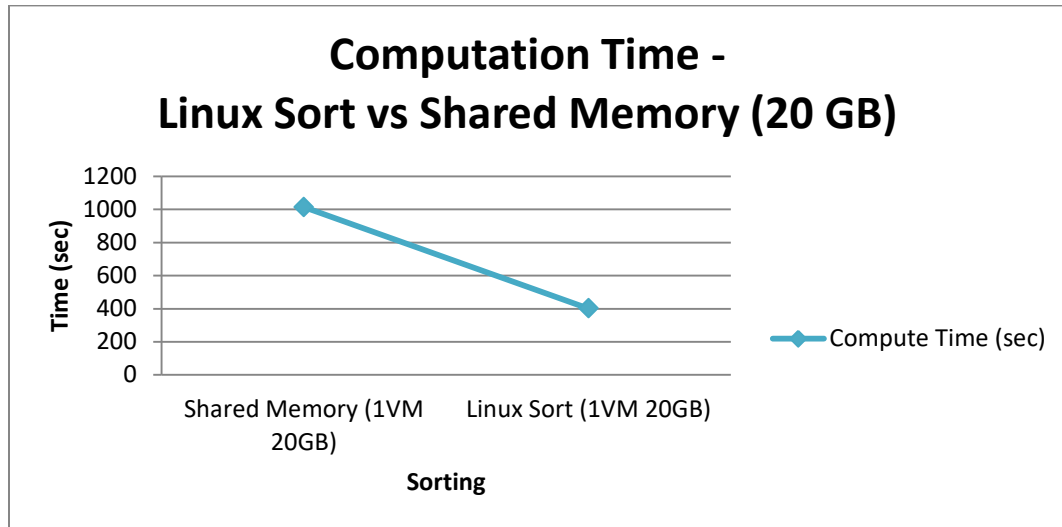


Fig 6 : Computation time – Linux Sort vs shared Memory – 20GB

- Fig 6 compares the compute time for linux sort and shared memory for 20GB file. Here too, the time taken for linux sort on 20 GB file was almost 611 seconds less than the sorting I performed and the reason being same as linux sort uses efficient algorithm to sort the records by using more threads and also depends in the language used etc

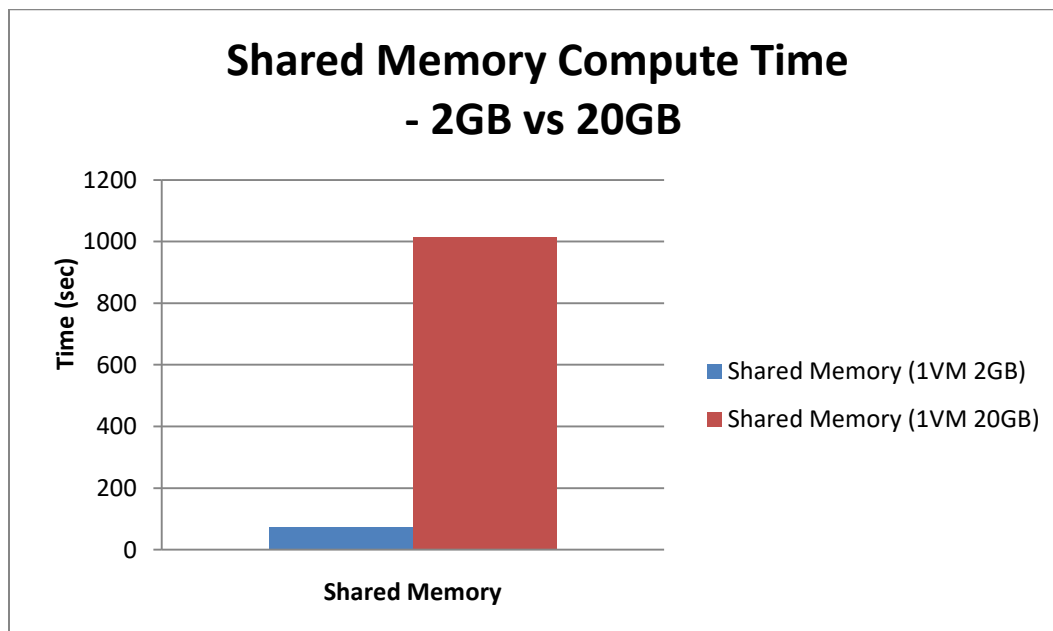


Fig 7 : Shared memory Compute time - 2GB vs 20GB

- Fig 7 compares the compute time for shared memory for 2GB file and 20 GB file. The time taken for linux sort on 2 GB file was 73.629 sec and on 20 GB file was 1013.88 sec as here the number of records were less in 2GB file than 20GB file

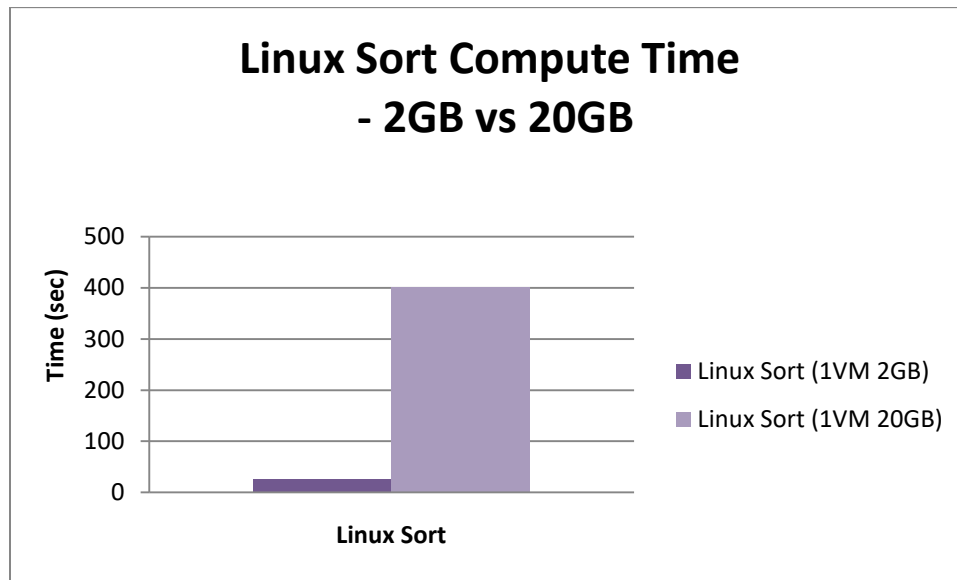


Fig 8 : Linux sort Compute time - 2GB vs 20GB

- Fig 8 compares the compute time for linux sort for 2GB file and 20 GB file. The time taken for linux sort on 2 GB file was 25 sec and on 20 GB file was 402 sec as the number of records were less in 2GB file than 20GB file so 2GB got sorted faster as the workload is less

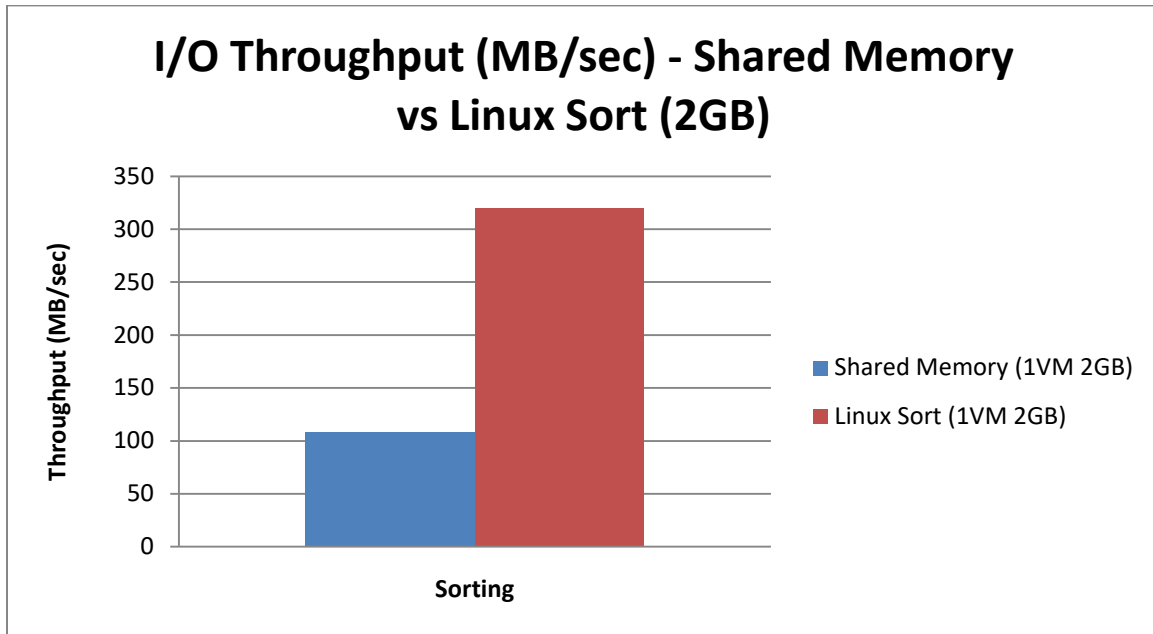


Fig 9 : I/O Throughput (MB/sec) – Shared memory vs Linux sort for 2GB

- Fig 9 compares the I/O throughput in MB/sec for shared memory and linux sort for 2 GB file. The linux sort has throughput of 320 MB/sec and mysort has 108.65 MB/sec. The reason that linux sort has better throughput than mysort is that it takes less time to sort the 2GB file and also uses efficient sorting algorithm with multiple threads

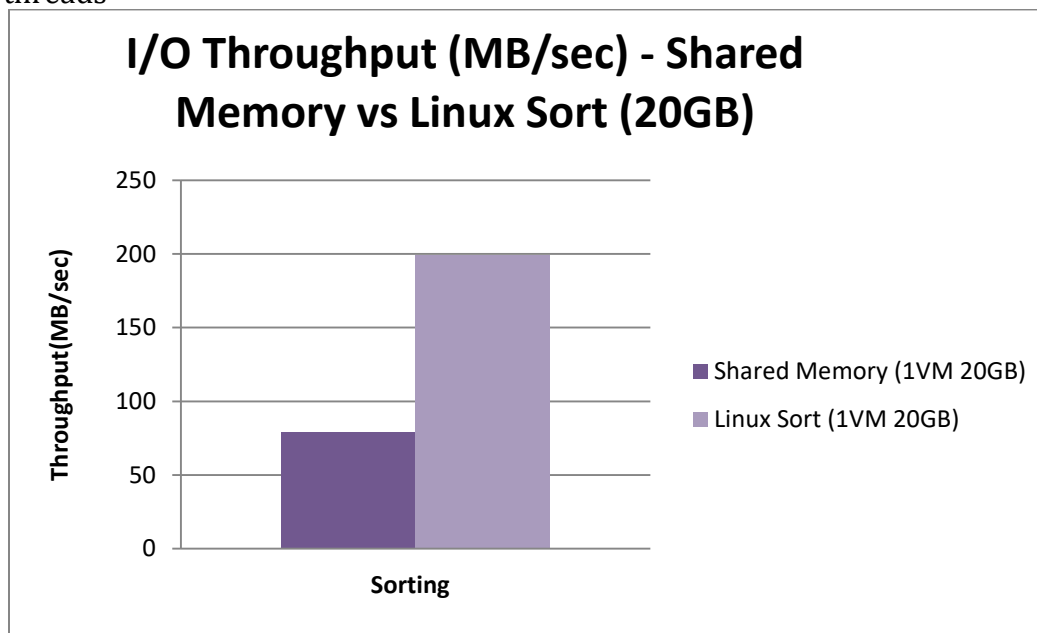


Fig 10 : I/O Throughput (MB/sec) – Shared memory vs Linux sort for 20 GB

- Fig 10 compares the I/O throughput in MB/sec for shared memory and linux sort for 20 GB file. The linux sort has throughput of 199 MB/sec and mysort has 78.90 MB/sec. The same story, linux sort has better throughput than mysort because it takes less time to sort the 20 GB file than the time taken for mysort to sort the 20 GB file and also uses efficient sorting algorithm with multiple threads

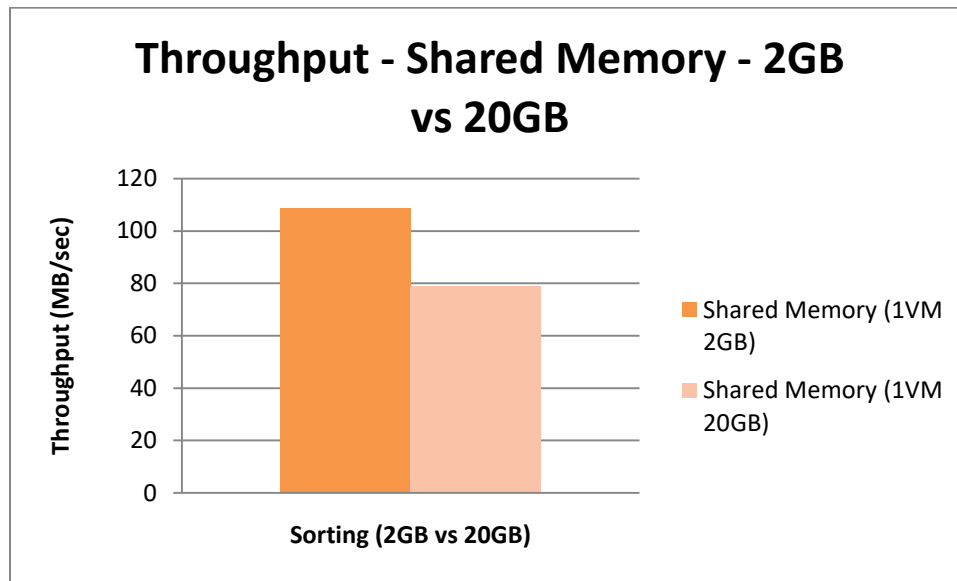


Fig 11 : Throughput - Shared Memory - 2GB vs 20GB

- Fig 11 compares the I/O throughput in MB/sec for shared memory 2GB vs 20 GB file. The throughput for 2GB file is 108.65 MB/sec and for 20 GB file is 78.90 MB/sec. As the workload is less for 2GB file i.e. it has less number of records than 20 GB file's records hence it has better throughput than the one for 20 GB file

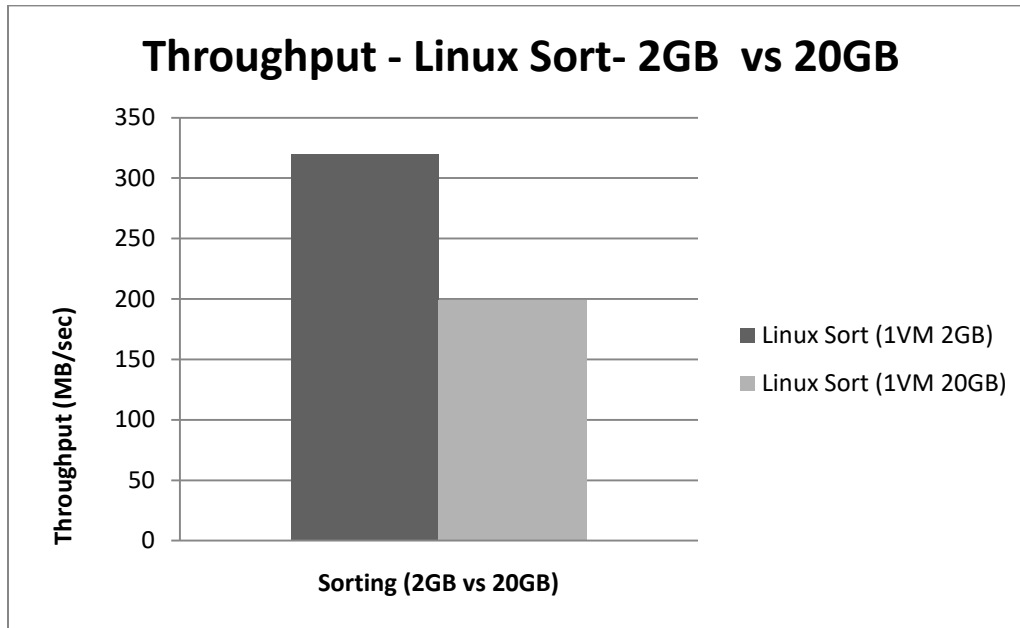


Fig 12 : Throughput -Linux sorts - 2GB vs 20GB

- Fig 12 compares the I/O throughput in MB/sec for linux 2GB vs 20 GB file. The throughput for 2GB file is 320 MB/sec and for 20 GB file is 199 MB/sec. As the workload is less for 2GB file i.e. it has less number of records than 20 GB file's records hence it has better throughput than the one for 20 GB file

IMPROVEMENTS / FUTURE WORK:

- This can have future work or improvements by performing sorting using more threads and by implementing efficient approach than the one used for mysort