Cloud Computing
HW5

Amrutham Lakshmi Himaja

A20474105

Report.txt

In this assignment the input file is the file which is generated by gensort to sort the data.If the file size is less than 8gb, quick sort is executed within the memory because the file size is less than the memory size(8gb). In this code, for internal sort I have used 2 and 4 threads both for Shared memory program and linux sort utility.

If the file size is greater than 8gb, external sorting is used because the file size is greater than memory size(8gb) and the total file cannot be loaded into memory at once. So the input file is divided into chunks of size 8gb which can be sorted making fit into the memory.In every iteration 8gb of data is sorted and written into the temp file. All the chunks are merged to the final sorted output. This way keep removing nodes and adding nodes until we finish all the chunk files and push the data to the output file.For external sort,I have used 24 and 48 threads both for Shared memory program and linux sort utility.

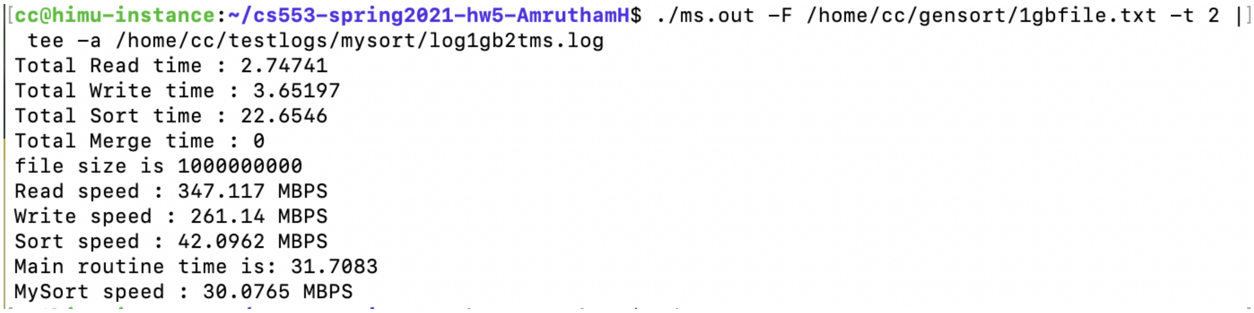
Using the output of MySort.cpp the log files are created which contains write time,read time,sort time data.
The folder **saroutputs** have the sar outputs of multiple sizes and multiple threads and the test folder has all the all of multiple file sizes and multiple threads.The **logs folder** has the log files of the best combination of file size and threads used.
For linsort for all the file sizes i got an issue in generating log files so i have attached the corresponding screenshots for 1gb,4gb and 16gb file sizes.For the file size 64gb i was not able to get the output for the time command so i have the sar file as a justification for 24 threads and 48threads respectively.

Experiment	**Shared Memory (1GB)**	**Linux Sort (1GB)**	**Shared Memory (4GB)**	**Linux Sort (4GB)**	**Shared Memory (16GB)**	**Linux Sort (16GB)**	**Shared Memory (64GB)**	**Linux Sort (64GB)**
Number of Threads	2	4	4	4	48	48	48	48
Sort Approach (e.g. in-memory / external)	In-memory	In-memory	In-memory	In-memory	External memory	External memory	External memory	External memory
Sort Algorithm (e.g. quicksort / mergesort / etc)	Quick Sort	Merge Sort	Quick Sort	Merge Sort	Quick Sort	Merge Sort	Quick Sort	Merge Sort
Data Read (GB)	1	1	4	8.922	32	31.656	128	127.993
Data Write (GB)	1	1	4	7.233	32	29.947	128	164.001
Sort Time (sec)	31.708	13.298	152.067	54.057	227.595	191.427	993.922	1020.762
Overall I/O Throughput (MB/sec)	278.34	247.81	107.79	219.81	530.01	583.96	435.83	312.49
Overall CPU	1.7	4.25	2.17	4.37	6.98	10.94	5.54	4.43

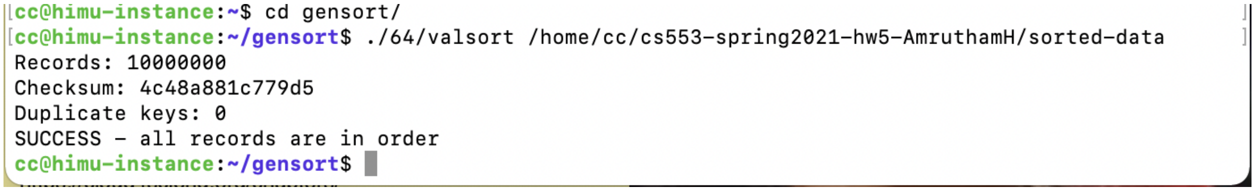
Utilization (%)								
Average Memory Utilization (GB)	1.540	1.720	5.166	5.086	5.275	7.266	4.259	4.297

According to my observations from the table we can see that MySort.cpp performs better than linux sort for files 16gb and 64gb.For 1gb and 4gb linux sort performance is better than shared memory.I/O throughput utilization is above 200 MB/s for most of the files,Overall CPU utilization for external sort is quite higher.
There are 48 logical cores on the skylake machine and hence one thread can run one logical core which results in best performance.

MySort on 1gb file:

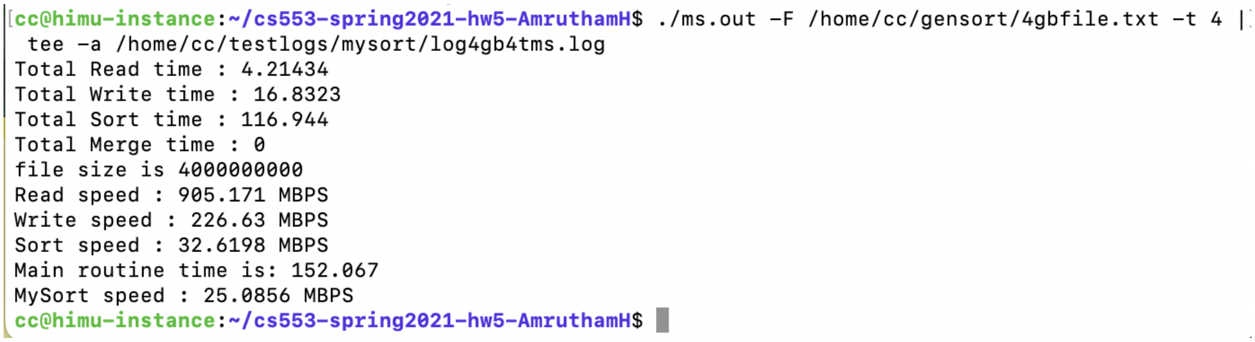


```
cc@himu-instance:~/cs553-spring2021-hw5-AmruthamH$ ./ms.out -F /home/cc/gensort/1gbfile.txt -t 2 |
 tee -a /home/cc/testlogs/mysort/log1gb2tms.log
Total Read time : 2.74741
Total Write time : 3.65197
Total Sort time : 22.6546
Total Merge time : 0
file size is 1000000000
Read speed : 347.117 MBPS
Write speed : 261.14 MBPS
Sort speed : 42.0962 MBPS
Main routine time is: 31.7083
MySort speed : 30.0765 MBPS
```

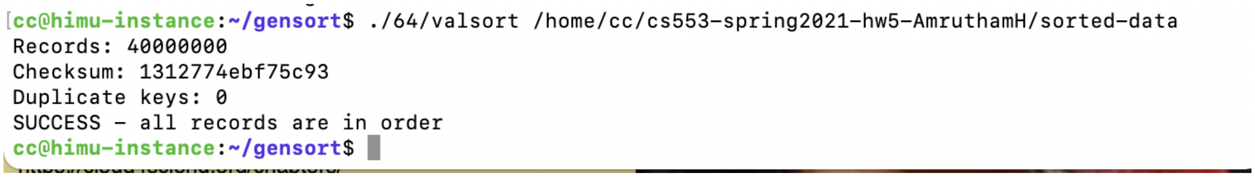


```
cc@himu-instance:~$ cd gensort/
cc@himu-instance:~/gensort$ ./64/valsort /home/cc/cs553-spring2021-hw5-AmruthamH/sorted-data
Records: 10000000
Checksum: 4c48a881c779d5
Duplicate keys: 0
SUCCESS - all records are in order
cc@himu-instance:~/gensort$
```

MySort on 4gb file:

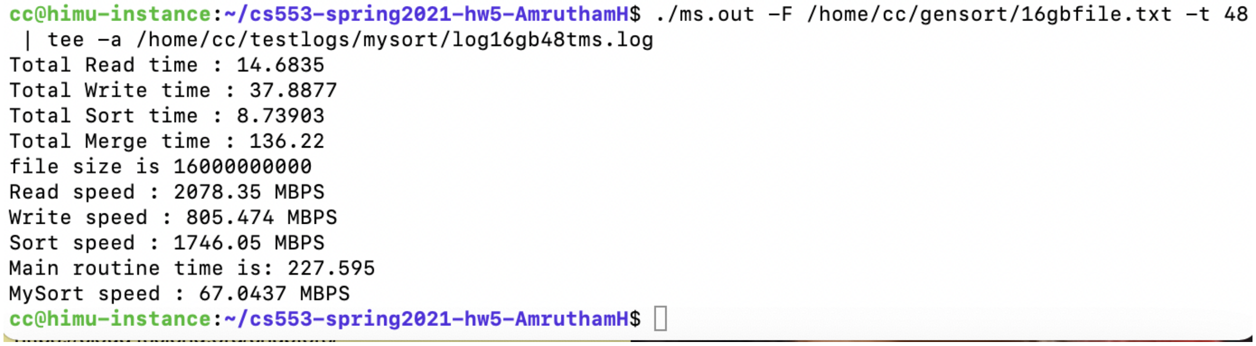


```
cc@himu-instance:~/cs553-spring2021-hw5-AmruthamH$ ./ms.out -F /home/cc/gensort/4gbfile.txt -t 4 |
 tee -a /home/cc/testlogs/mysort/log4gb4tms.log
Total Read time : 4.21434
Total Write time : 16.8323
Total Sort time : 116.944
Total Merge time : 0
file size is 4000000000
Read speed : 905.171 MBPS
Write speed : 226.63 MBPS
Sort speed : 32.6198 MBPS
Main routine time is: 152.067
MySort speed : 25.0856 MBPS
cc@himu-instance:~/cs553-spring2021-hw5-AmruthamH$
```

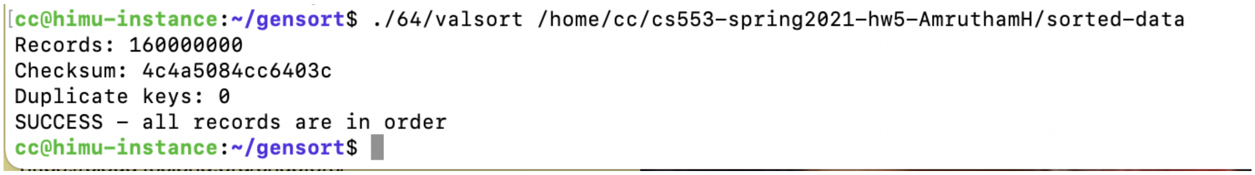


```
cc@himu-instance:~/gensort$ ./64/valsort /home/cc/cs553-spring2021-hw5-AmruthamH/sorted-data
Records: 40000000
Checksum: 1312774ebf75c93
Duplicate keys: 0
SUCCESS - all records are in order
cc@himu-instance:~/gensort$
```

MySort on 16gb file:

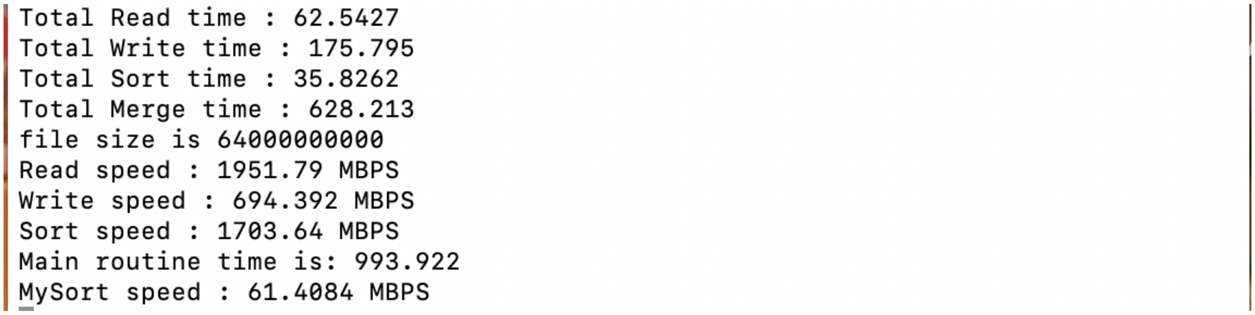


```
cc@himu-instance:~/cs553-spring2021-hw5-AmruthamH$ ./ms.out -F /home/cc/gensort/16gbfile.txt -t 48
 | tee -a /home/cc/testlogs/mysort/log16gb48tms.log
Total Read time : 14.6835
Total Write time : 37.8877
Total Sort time : 8.73903
Total Merge time : 136.22
file size is 16000000000
Read speed : 2078.35 MBPS
Write speed : 805.474 MBPS
Sort speed : 1746.05 MBPS
Main routine time is: 227.595
MySort speed : 67.0437 MBPS
cc@himu-instance:~/cs553-spring2021-hw5-AmruthamH$
```

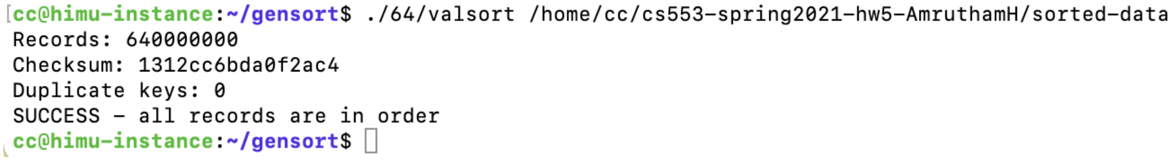


```
cc@himu-instance:~/gensort$ ./64/valsort /home/cc/cs553-spring2021-hw5-AmruthamH/sorted-data
Records: 160000000
Checksum: 4c4a5084cc6403c
Duplicate keys: 0
SUCCESS - all records are in order
cc@himu-instance:~/gensort$
```

MySort on 64gb file:



```
Total Read time : 62.5427
Total Write time : 175.795
Total Sort time : 35.8262
Total Merge time : 628.213
file size is 64000000000
Read speed : 1951.79 MBPS
Write speed : 694.392 MBPS
Sort speed : 1703.64 MBPS
Main routine time is: 993.922
MySort speed : 61.4084 MBPS
```



```
cc@himu-instance:~/gensort$ ./64/valsort /home/cc/cs553-spring2021-hw5-AmruthamH/sorted-data
Records: 640000000
Checksum: 1312cc6bda0f2ac4
Duplicate keys: 0
SUCCESS - all records are in order
cc@himu-instance:~/gensort$
```

Note: I am submitting this assignment after one day of deadline because i was using the instance which was created by my friend on the last day the Chameleon cloud lease ended when pushing the data to github repository. I would kindly like to request you to grade my assignment accordingly.