CS553 Homework #5

Sort on Single Shared Memory Node

Bhavya Chawla (bchawla@hawk.iit.edu) A2051695

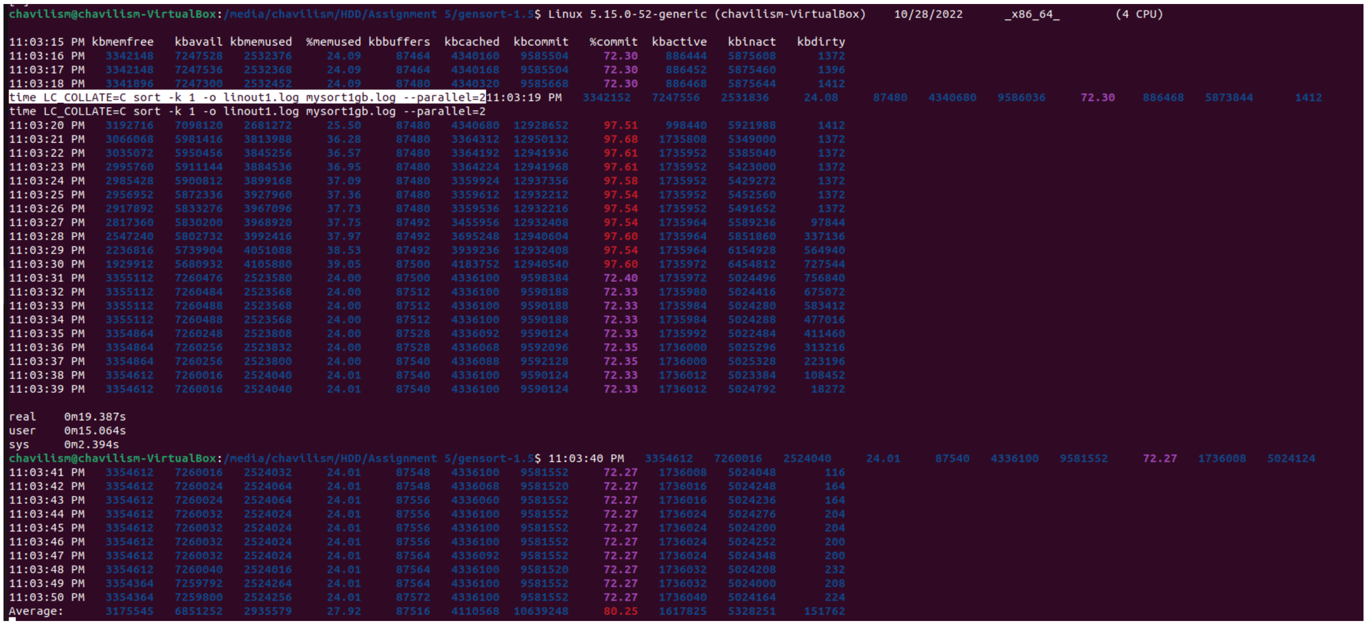
Introduction

The goal of this programming assignment is to enable you to gain experience programming with external data sort and multi-threaded programming.

Table 1: Performance evaluation of Single Node TeraSort (using best # of threads for each case)

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	2	2	2	2	2	2	2
Sort Approach (e.g. in-memory / external)	In-Memory	In-Memory	External	External	External	External	External	External
Sort Algorithm (e.g. quicksort / mergesort / etc)	Mergesort	MergeSort	Mergesort	MergeSort	Mergesort	Mergesort	Mergesort	Mergesort
Data Read (GB)	1	1	8	8	32	32	128	128
Data Write (GB)	1	1	8	8	32	32	128	128
Sort Time (sec)	18.79	19.387	295.56	118	1945.78	523	6733.52	2306
Overall I/O Throughput (MB/sec)	106.43	103.16	54.13	135.59	32.89	122.37	38.01	111.01
Overall CPU Utilization (%)	37.45	27.09	43.52	31.95	48.83	37.41	50.66	42.26
Average Memory Utilization (GB)	2.9	2.1	3.4	2.56	3.9	2.99	4.0	3.3

For Linux Sort, Used **SAR** cmd for %memused - Avg memory used, and **LC_COLLATE=C** for sorting special characters as otherwise Sort cmd does not work with special characters.

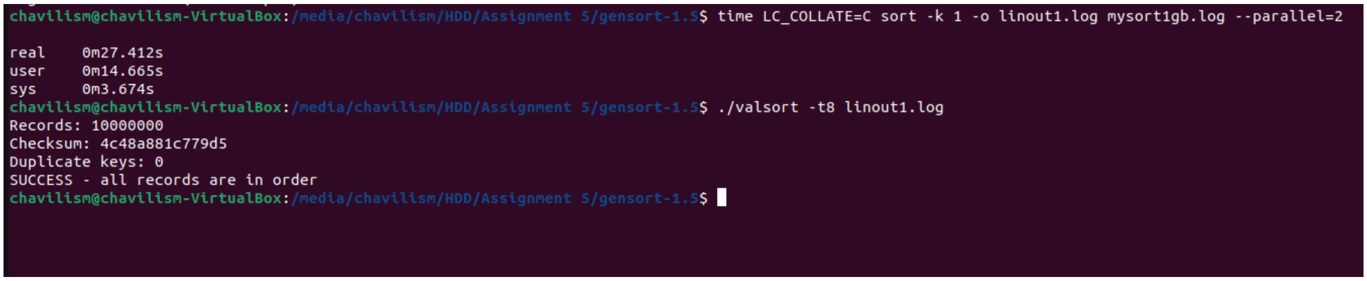


```
chavilism@chavilism-VirtualBox:/media/chavilism/HDD/Assignment 5/gensort-1.5$ Linux 5.15.0-52-generic (chavilism-VirtualBox)    10/28/2022      _x86_64_        (4 CPU)

11:03:15 PM kbmemfree   kbavail kbmemused  %memused kbbuffers  kbcached  kbcommit   %commit  kbactive   kbinact   kbdirty
11:03:16 PM   3342148   7247528   2532376     24.09     87464   4340160   9585504     72.30    886444   5875608      1372
11:03:17 PM   3342148   7247536   2532368     24.09     87464   4340168   9585504     72.30    886452   5875460      1396
11:03:18 PM   3341896   7247300   2532452     24.09     87480   4340320   9585668     72.30    886468   5875644      1412
time LC_COLLATE=C sort -k 1 -o linout1.log mysort1gb.log --parallel=211:03:19 PM   3342152   7247556   2531836     24.08     87480   4340680   9586036     72.30    886468   5873844      1412
time LC_COLLATE=C sort -k 1 -o linout1.log mysort1gb.log --parallel=2
11:03:20 PM   3192716   7098120   2681272     25.50     87480   4340680  12928652     97.51    998440   5921988      1412
11:03:21 PM   3066068   5981416   3813988     36.28     87480   3364312  12950132     97.68   1735808   5349000      1372
11:03:22 PM   3035072   5950456   3845256     36.57     87480   3364192  12941936     97.61   1735952   5385040      1372
11:03:23 PM   2995760   5911144   3884536     36.95     87480   3364224  12941968     97.61   1735952   5423000      1372
11:03:24 PM   2985428   5900812   3899168     37.09     87480   3359924  12937356     97.58   1735952   5429272      1372
11:03:25 PM   2956952   5872336   3927960     37.36     87480   3359612  12932212     97.54   1735952   5452560      1372
11:03:26 PM   2917892   5833276   3967096     37.73     87480   3359536  12932216     97.54   1735952   5491652      1372
11:03:27 PM   2817360   5830200   3968920     37.75     87492   3455956  12932408     97.54   1735964   5589236     97844
11:03:28 PM   2547240   5802732   3992416     37.97     87492   3695248  12940604     97.60   1735964   5851860    337136
11:03:29 PM   2236816   5739904   4051088     38.53     87492   3939236  12932408     97.54   1735964   6154928    564940
11:03:30 PM   1929912   5680932   4105880     39.05     87500   4183752  12940540     97.60   1735972   6454812    727544
11:03:31 PM   3355112   7260476   2523580     24.00     87500   4336100   9598384     72.40   1735972   5024496    756840
11:03:32 PM   3355112   7260484   2523568     24.00     87512   4336100   9590188     72.33   1735980   5024416    675072
11:03:33 PM   3355112   7260488   2523568     24.00     87512   4336100   9590188     72.33   1735984   5024280    583412
11:03:34 PM   3355112   7260488   2523568     24.00     87512   4336100   9590188     72.33   1735984   5024288    477016
11:03:35 PM   3354864   7260248   2523808     24.00     87528   4336092   9590124     72.33   1735992   5022484    411460
11:03:36 PM   3354864   7260256   2523832     24.00     87528   4336068   9592096     72.35   1736000   5025296    313216
11:03:37 PM   3354864   7260256   2523800     24.00     87540   4336088   9592128     72.35   1736000   5025328    223196
11:03:38 PM   3354612   7260016   2524040     24.01     87540   4336100   9590124     72.33   1736012   5023384    108452
11:03:39 PM   3354612   7260016   2524040     24.01     87540   4336100   9590124     72.33   1736012   5024792     18272

real    0m19.387s
user    0m15.064s
sys     0m2.394s
chavilism@chavilism-VirtualBox:/media/chavilism/HDD/Assignment 5/gensort-1.5$ 11:03:40 PM   3354612   7260016   2524040     24.01     87540   4336100   9581552     72.27   1736008   5024124
11:03:41 PM   3354612   7260016   2524032     24.01     87548   4336100   9581552     72.27   1736008   5024048       116
11:03:42 PM   3354612   7260024   2524064     24.01     87548   4336068   9581520     72.27   1736016   5024248       164
11:03:43 PM   3354612   7260024   2524064     24.01     87556   4336060   9581552     72.27   1736016   5024236       164
11:03:44 PM   3354612   7260032   2524024     24.01     87556   4336100   9581552     72.27   1736024   5024276       204
11:03:45 PM   3354612   7260032   2524024     24.01     87556   4336100   9581552     72.27   1736024   5024200       204
11:03:46 PM   3354612   7260032   2524024     24.01     87556   4336100   9581552     72.27   1736024   5024252       200
11:03:47 PM   3354612   7260032   2524024     24.01     87564   4336092   9581552     72.27   1736024   5024348       200
11:03:48 PM   3354612   7260040   2524016     24.01     87564   4336100   9581520     72.27   1736032   5024208       232
11:03:49 PM   3354364   7259792   2524264     24.01     87564   4336100   9581552     72.27   1736032   5024000       208
11:03:50 PM   3354364   7259800   2524256     24.01     87572   4336100   9581552     72.27   1736040   5024164       224
Average:      3175545   6851252   2935579     27.92     87516   4110568  10639248     80.25   1617825   5328251    151762
```

using valsort to check for order, in the output files,



```
chavilism@chavilism-VirtualBox:/media/chavilism/HDD/Assignment 5/gensort-1.5$ time LC_COLLATE=C sort -k 1 -o linout1.log mysort1gb.log --parallel=2

real    0m27.412s
user    0m14.665s
sys     0m3.674s
chavilism@chavilism-VirtualBox:/media/chavilism/HDD/Assignment 5/gensort-1.5$ ./valsort -t8 linout1.log
Records: 10000000
Checksum: 4c48a881c779d5
Duplicate keys: 0
SUCCESS - all records are in order
chavilism@chavilism-VirtualBox:/media/chavilism/HDD/Assignment 5/gensort-1.5$ 
```

Calculations:

1. Overall I/O is calculated by (Data Read(MB) + Data Write(MB))/Sorting time(s)
2. Average Memory Utilization is calculated by considering 8GB (8000MB) memory allocated for the virtual machine and then per the %memused/CPU Utilization,
   i.e 37.45% of 8000MB
3. Data Read and Data Write is depended upon sort type, In-memory or External sort,
   for external sort per the algorithm, we are reading and writing twice the amount of data to/from storage.

   For external sort k-way Merge Sort is better than other sorting techniques as with the avg. complexity O(nlgn)

1. What you will submit

2.

The grading will be done according to the rubric below:

- Shared memory sort implementation/scripts: 50 points
- README.md: 5 points
- Performance evaluation, data, explanations, etc: 40 points
- Followed instructions on deliverables: 5 points

files:

- Makefile
- mysort.c
- hw5_report.pdf
- README.md

- mysort1GB.log
- mysort4GB.log
- mysort16GB.log
- mysort64GB.log

- linsort1GB.log
- linsort4GB.log
- linsort16GB.log
- linsort64GB.log