CS553 Cloud computing - Homework 4

Name: **Batkhishig** Dulamsurankhor
CWID: A20543498

Hardware information

- CPU: 32x Intel(R) Xeon(R) Silver 4108 CPU @ 1.80GHz
- Memory: 64GB
- Disk: Micron_5100_MTFD 240GB

Test results

I have written a **run.sh** script to automate the testing and saving the results to files. After running the benchmark and collecting them in the host machine, we can use **plot.py** script to plot graphs for each of the benchmarks (cpu, memory, disk and network) from the raw log files. The instruction to use these scripts is in the README.md file.

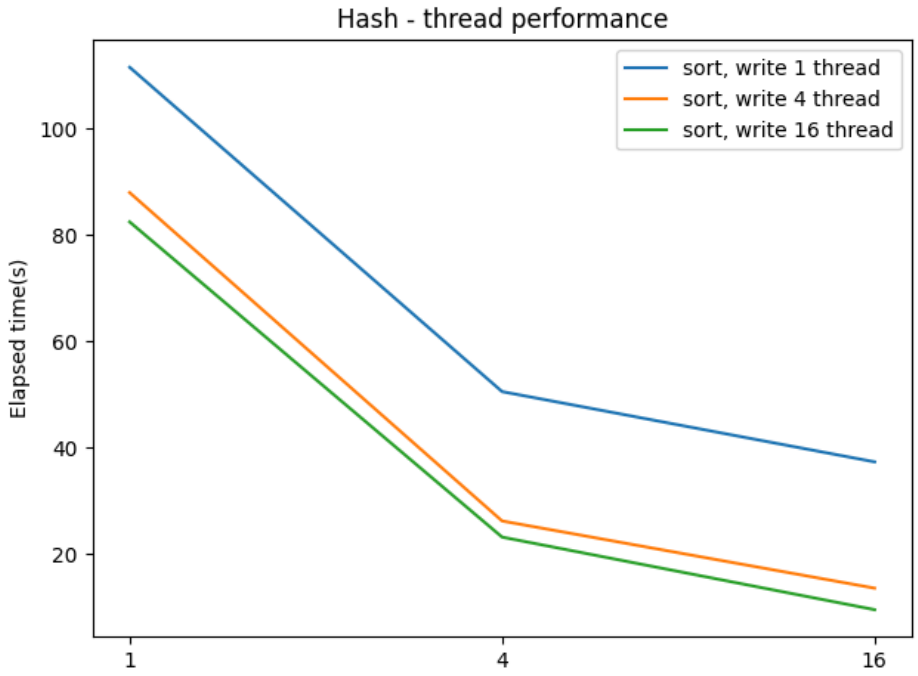
CPU benchmark

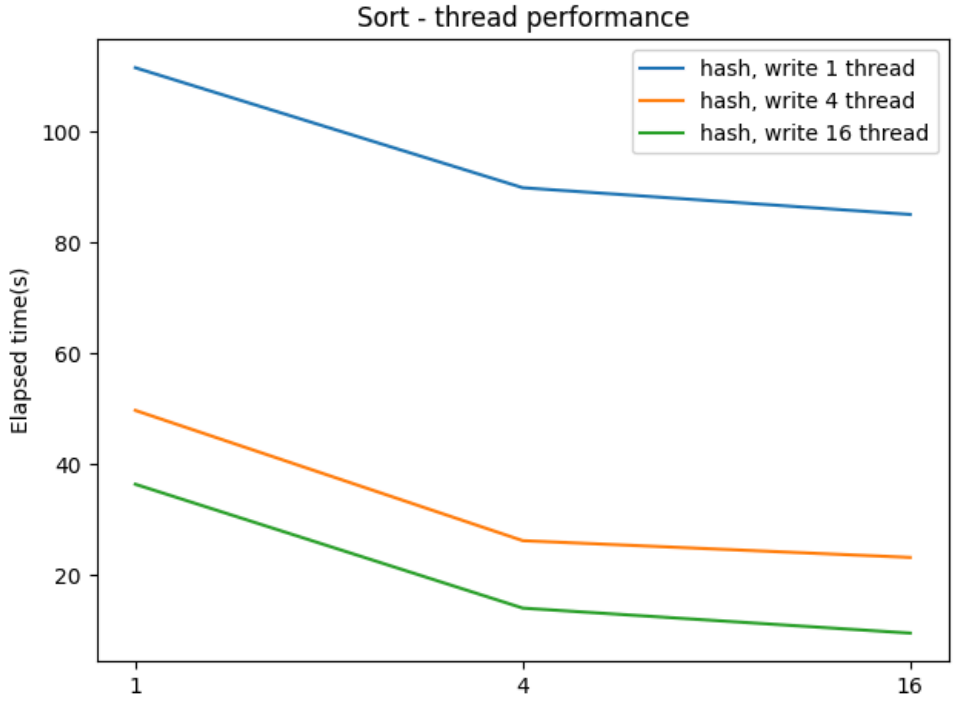
Following is the test result for 1GB small file for variable thread configurations for hashing, sorting and writing to a file. The performance gain is calculated by 100-elapsed_time/(1,1,1 thread elapsed time)*100. Relative performance to the test done with 1 thread for each stage of the test.

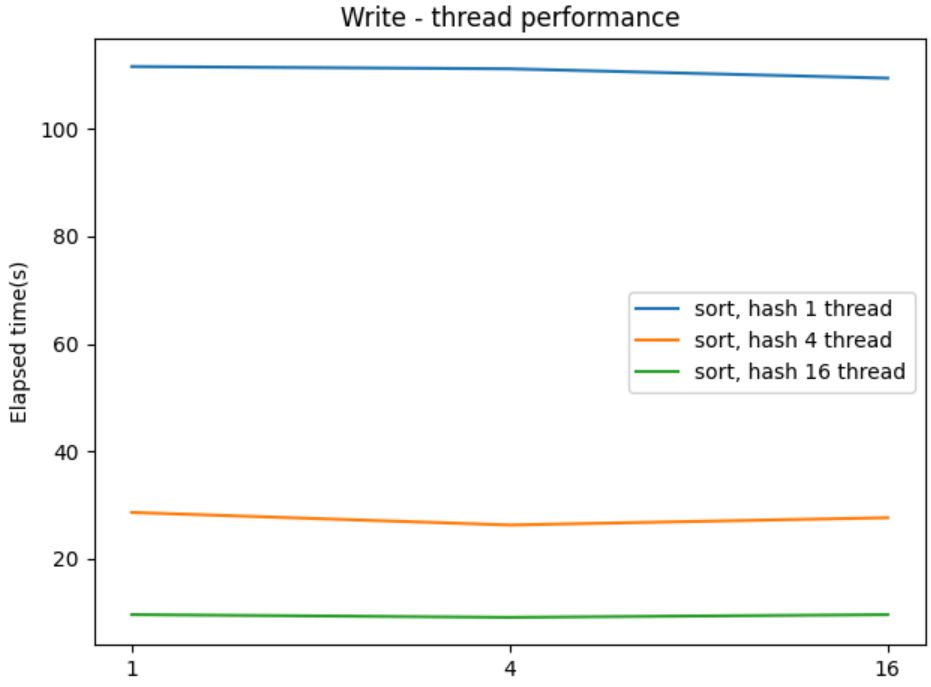
Hash thread count	Sort thread count	Write thread count	Total elapsed time(s)	Performance boost
1	1	1	111.598420	0.00%
1	1	4	111.177812	0.38%
1	1	16	109.433756	1.94%
1	4	1	89.949533	19.40%
1	4	4	88.025533	21.12%
1	4	16	87.847423	21.28%
1	16	1	85.129884	23.72%
1	16	4	82.648107	25.94%
1	16	16	82.513461	26.06%
4	1	1	50.602234	54.66%
4	1	4	49.776363	55.40%
4	1	16	49.780005	55.39%
4	4	1	28.614621	74.36%
4	4	4	26.277282	76.45%
4	4	16	27.633732	75.24%
4	16	1	23.084791	79.31%
4	16	4	23.250475	79.17%
4	16	16	23.244591	79.17%
16	1	1	37.395826	66.49%
16	1	4	36.355394	67.42%
16	1	16	36.453942	67.33%
16	4	1	14.744985	86.79%
16	4	4	13.647057	87.77%
16	4	16	14.108865	87.36%
16	16	1	9.606335	91.39%
16	16	4	9.088045	91.86%
16	16	16	9.597904	91.40%

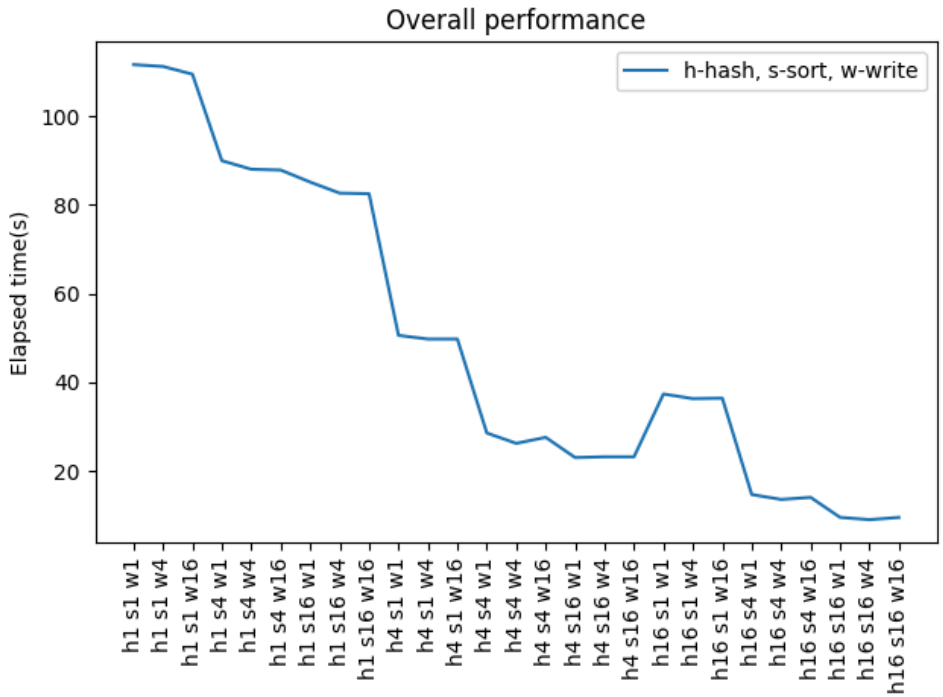
From the results, we can see that hash generation had the most benefit from multithreading, while writing to a file didn’t have much gain. Writing with 16 threads had more overheads than performance gain in some cases.

Graphs


Hash - thread performance
sort, write 1 thread
sort, write 4 thread
sort, write 16 thread
Elapsed time(s)
100
80
60
40
20
1
4
16


Sort - thread performance
hash, write 1 thread
hash, write 4 thread
hash, write 16 thread
Elapsed time(s)
100
80
60
40
20
1
4
16


Write - thread performance
Elapsed time(s)
100
80
60
40
20
1
4
16
sort, hash 1 thread
sort, hash 4 thread
sort, hash 16 thread


Overall performance
Elapsed time(s)
100
80
60
40
20
h-hash, s-sort, w-write
h1 s1 w1
h1 s1 w4
h1 s1 w16
h1 s4 w1
h1 s4 w4
h1 s4 w16
h1 s16 w1
h1 s16 w4
h1 s16 w16
h4 s1 w1
h4 s1 w4
h4 s1 w16
h4 s4 w1
h4 s4 w4
h4 s4 w16
h4 s16 w1
h4 s16 w4
h4 s16 w16
h16 s1 w1
h16 s1 w4
h16 s1 w16
h16 s4 w1
h16 s4 w4
h16 s4 w16
h16 s16 w1
h16 s16 w4
h16 s16 w16

Screenshot from run.sh: