Exercise 01

VU Performance-oriented Computing, Summer Semester 2024

Calvin Hoy

2024-03-12

Outline/Preparation

Test environment (LCC3)

Benchmarks were conducted on the LCC3 cluster only.

I loaded the `gcc/12.2.0-gcc-8.5.0-p4pe45v` module before building any of the code.

I have attached the Slurm job scripts I used to run the benchmarks on LCC3's compute nodes in the `jobs/` directory.

Updated benchmark scripts

The benchmark.sh script from exercise 1 has been updated to, by default, re-run the benchmark until a statistical error below 0.05 has been reached. This may be configured by the user; see `--help`.

bench_small_samples.sh has been amended to allow optionally running either the CPU load generator (supplying the path to the `tools` directory) or I/O load generator (supplying the path to the `ioloadgen/` directory. Furthermore, the maximum number of files created for the `filegen` and `filesearch` benchmarks has been reduced to 10,000.

I/O load generator

I chose to use C++, but using C file

The generator works by creating a working file of a certain size (8 GiB by default, configurable via an environment variable) and writing data to it.

The program can be run in two primary modes: `--sequential` and `--random`. In sequential mode, the working file is repeatedly overwritten start-to-finish in 1 MiB chunks, which primarily exercises the disk's sequential write performance. In random mode, the file will instead have random 512-byte sections , which is constrained primarily by the rate at which the disk can complete individiual I/O operations. In either mode, each block is flushed to disk immediately, in order to minimize the impact of write caching.

By default, the program will write as fast as the disk allows, essentially creating a 100% I/O load. It can be limited to a specified write rate (either in B/s or IOPS, depending on the mode) with `--limit <rate>`.

The program has a third mode, `--calib`, which can be used to measure disk performance. In this program, sequential and random tests are run for 10 seconds each, keeping track of the achieved output, then display the results in Bytes per second (B/s) and I/O Operations per second (IOPS) respectively.

On my personal laptop, I measured around 2.6 GiB/s for sequential write and 160,000 IOPS for random write.

You may find the code in `ioloadgen/ioloadgen.cpp`. See `--help` for additional information.

Benchmark results

All figures for mean and variance given in the following section were obtained by re-running the benchmark until a statistical error below 0.05 or reached or a maximum number of runs (15) was reached. Unless otherwise specified, runtime is specified in seconds and memory use in kilobytes.

Raw (JSON) output from each of the tests as performed on LCC3 can be found in `results/` directories.

All benchmarks were rerun for this exercise.

CPU load generator

I chose to measure the impact of the CPU load generator provided in the `tools` directory, executed using (a modified version of) the `exec_with_workstation_heavy.sh` script, on the CPU-limited programs provided in `small-samples`.

`delannoy`

No load

	Mean				Variance			
N	wall	user	system	mem	wall	user	system	mem
1	0.000	0.000	0.000	1350.667	0.000	0.000	0.000	1029.333
2	0.000	0.000	0.000	1361.333	0.000	0.000	0.000	1765.333
3	0.000	0.000	0.000	1338.667	0.000	0.000	0.000	709.333
4	0.000	0.000	0.000	1342.667	0.000	0.000	0.000	485.333
5	0.000	0.000	0.000	1356.000	0.000	0.000	0.000	1296.000
6	0.000	0.000	0.000	1340.000	0.000	0.000	0.000	208.000
7	0.000	0.000	0.000	1333.333	0.000	0.000	0.000	1557.333
8	0.000	0.000	0.000	1312.000	0.000	0.000	0.000	912.000
9	0.010	0.007	0.000	1354.667	0.000	0.000	0.000	3845.333
10	0.050	0.050	0.000	1356.000	0.000	0.000	0.000	0.000
11	0.307	0.307	0.000	1358.667	0.000	0.000	0.000	37.333
12	1.710	1.707	0.000	1362.667	0.000	0.000	0.000	1477.333
13	9.637	9.617	0.000	1333.333	0.000	0.000	0.000	357.333
14	54.265	54.182	0.000	1322.000	0.008	0.008	0.000	837.333

With load

	Mean				Variance			
N	wall	user	system	mem	wall	user	system	mem
1	0.017	0.003	0.007	3440.000	0.000	0.000	0.000	81264.000
2	0.013	0.000	0.000	3261.333	0.000	0.000	0.000	2501.333
3	0.017	0.000	0.007	3236.000	0.000	0.000	0.000	3504.000
4	0.013	0.000	0.003	3277.333	0.000	0.000	0.000	2725.333
5	0.013	0.000	0.010	3272.000	0.000	0.000	0.000	3184.000
6	0.017	0.000	0.003	3309.333	0.000	0.000	0.000	837.333
7	0.017	0.000	0.007	3280.000	0.000	0.000	0.000	768.000
8	0.017	0.000	0.007	3310.667	0.000	0.000	0.000	405.333
9	0.020	0.010	0.003	3220.000	0.000	0.000	0.000	9408.000
10	0.090	0.067	0.003	3294.667	0.000	0.000	0.000	2949.333
11	0.420	0.343	0.010	3248.000	0.000	0.000	0.000	4656.000
12	1.963	1.870	0.010	3249.333	0.002	0.001	0.000	741.333
13	11.699	11.151	0.010	3282.857	0.016	0.017	0.000	1934.476
14	64.340	61.456	0.029	3272.267	0.119	0.082	0.000	2106.210

`mmul`

	wall	user	system	mem
No load - Mean	5.750	5.730	0.000	24622.667
No load - Variance	0.000	0.000	0.000	1125.333
With load - Mean	7.000	6.660	0.013	24533.333
With load - Variance	0.004	0.001	0.000	597.333

`nbody`

	wall	user	system	mem
No load - Mean	4.770	4.763	0.000	1870.667
No load - Variance	0.000	0.000	0.000	5.333
With load - Mean	5.717	5.450	0.010	3234.667
With load - Variance	0.005	0.001	0.000	5717.333

`qap`

No load

	Mean				**Variance**			
Input	wall	user	system	mem	wall	user	system	mem
chr10a	0.010	0.007	0.000	1512.000	0.000	0.000	0.000	784.000
chr12a	0.160	0.157	0.000	1506.667	0.000	0.000	0.000	709.333
chr12b	0.143	0.140	0.000	1497.333	0.000	0.000	0.000	69.333
chr12c	0.210	0.210	0.000	1505.333	0.000	0.000	0.000	197.333
chr15a	15.353	15.327	0.000	1502.667	0.001	0.002	0.000	517.333
chr15b	4.157	4.147	0.000	1500.000	0.000	0.000	0.000	112.000
chr15c	13.913	13.890	0.000	1512.000	0.000	0.000	0.000	784.000

With load

	Mean				**Variance**			
Input	wall	user	system	mem	wall	user	system	mem
chr10a	0.037	0.010	0.010	3278.667	0.000	0.000	0.000	3397.333
chr12a	0.213	0.177	0.007	3272.000	0.000	0.000	0.000	624.000
chr12b	0.230	0.163	0.010	3297.333	0.000	0.000	0.000	1941.333
chr12c	0.307	0.233	0.010	3285.333	0.000	0.000	0.000	869.333
chr15a	17.940	17.073	0.013	3292.000	0.001	0.001	0.000	400.000
chr15b	4.927	4.663	0.010	3238.667	0.001	0.000	0.000	1621.333
chr15c	16.440	15.653	0.010	3248.000	0.002	0.001	0.000	144.000

I/O load generator

LCC3 turned out to be problematic in that I could not seem to calibrate against its /tmp file system; the reported figures (>= 1 GiB/s, >= 300k IOPS) make **NO** sense whatsoever for a spinning hard disk.

Due to time constraints, I decided to assume a sequential write rate of 100MB/s and ran benchmarks with an external I/O load of 0%, 10%, 50% and 100% of that.

`filegen`

Unfortunately, no result within a statistical error boundary of 5% could be obtained for the other test parameters used in exercise sheet 1.

I/O load				**Mean**				**Variance**			
	dirs	files	file size	wall	user	system	mem	wall	user	system	m
-	1000	1	1	11.744	0.005	0.101	1478.000	7.769	0.000	0.000	1121
10 MiB/s	1	1	10000	0.000	0.000	0.000	1462.667	0.000	0.000	0.000	229
10 MiB/s	1	1	100000	0.030	0.000	0.000	1570.667	0.001	0.000	0.000	69
10 MiB/s	1	1	1000000	0.020	0.010	0.000	2057.333	0.000	0.000	0.000	1045

I/O load				Mean				Variance			
	dirs	files	file size	wall	user	system	mem	wall	user	system	
10 MiB/s	1	1	10000000	0.360	0.140	0.004	10976.800	0.019	0.000	0.000	16027
10 MiB/s	1	1	100000000	3.058	1.426	0.080	98972.800	0.792	0.000	0.000	1987
10 MiB/s	1	1000	1	9.926	0.004	0.108	1494.400	42.121	0.000	0.003	1356
10 MiB/s	1	10000	1	86.446	0.074	1.292	1459.200	2341.854	0.001	0.262	5747
10 MiB/s	1000	1	1	11.262	0.006	0.114	1480.800	52.729	0.000	0.002	1883
10 MiB/s	10000	1	1	122.602	0.098	1.324	1495.200	5613.920	0.001	0.192	1851
50 MiB/s	1	1000	1	10.566	0.004	0.124	1495.200	42.930	0.000	0.005	1579
50 MiB/s	1000	1	1	18.826	0.002	0.126	1480.000	115.709	0.000	0.002	1120
50 MiB/s	10000	1	1	168.536	0.092	1.306	1457.600	10458.891	0.001	0.176	4476
100 MiB/s	1	1000	1	12.368	0.000	0.118	1506.400	48.780	0.000	0.005	660
100 MiB/s	1000	1	1	14.160	0.004	0.126	1494.400	66.062	0.000	0.003	372
100 MiB/s	10000	1	1	159.648	0.086	1.204	1469.600	7959.468	0.001	0.137	380

`filesearch`

I/O load			Mean				Variance			
	dirs	files	wall	user	system	mem	wall	user	system	mem
-	1	100	0.000	0.000	0.000	1458.667	0.000	0.000	0.000	3621.333
-	1	10000	0.010	0.000	0.010	1409.333	0.000	0.000	0.000	5477.333
-	100	1	0.000	0.000	0.000	1416.000	0.000	0.000	0.000	624.000
-	100	100	0.010	0.000	0.010	1440.000	0.000	0.000	0.000	448.000
-	10000	1	0.080	0.013	0.063	1556.000	0.000	0.000	0.000	48.000
-	10000	1	0.080	0.010	0.067	1489.333	0.000	0.000	0.000	4181.333
10 MiB/s	1	100	0.000	0.000	0.000	1457.333	0.000	0.000	0.000	5541.333
10 MiB/s	1	10000	0.010	0.000	0.010	1454.667	0.000	0.000	0.000	2021.333
10 MiB/s	100	1	0.000	0.000	0.000	1436.000	0.000	0.000	0.000	336.000
10 MiB/s	100	100	0.010	0.000	0.010	1445.333	0.000	0.000	0.000	741.333
10 MiB/s	10000	1	0.083	0.010	0.067	1536.000	0.000	0.000	0.000	2032.000
50 MiB/s	1	100	0.000	0.000	0.000	1466.667	0.000	0.000	0.000	4501.333
50 MiB/s	1	10000	0.010	0.000	0.010	1486.667	0.000	0.000	0.000	4037.333
50 MiB/s	100	1	0.000	0.000	0.000	1453.333	0.000	0.000	0.000	5957.333
50 MiB/s	100	100	0.010	0.000	0.010	1438.667	0.000	0.000	0.000	341.333
50 MiB/s	10000	1	0.080	0.010	0.067	1428.000	0.000	0.000	0.000	768.000
100 MiB/s	1	100	0.000	0.000	0.000	1420.000	0.000	0.000	0.000	15424.000
100 MiB/s	1	10000	0.010	0.000	0.010	1365.333	0.000	0.000	0.000	6165.333
100 MiB/s	100	1	0.000	0.000	0.000	1458.667	0.000	0.000	0.000	485.333
100 MiB/s	100	100	0.010	0.000	0.010	1490.667	0.000	0.000	0.000	1605.333
100 MiB/s	10000	1	0.080	0.010	0.067	1465.333	0.000	0.000	0.000	917.333