Lab 07: Storage

The exercises marked with * are obligatory to complete before the coming weeks lecture.

The purpose of this lab is to introduce the operations of read and write to storage of your system, and to strengthen the understanding of read and write speeds. Additionally it challenges the student to implement a simple filesystem, which is rather complex.

Remember to write useful commands in your cheat sheets.

Table of content

Exercise 01*

In this exercise the student is going to run a read performance test of their storage units (HDD, SSD), using the tool fio.

Fio

The Flexible I/O Tester, `fio`, is a tool made for benchmarking I/O performance. When run, the fio tool will spawn a number of threads and do a user-selected I/O action. The action is specified by the user in a job file.

The `fio` tool is installed through the package manager. The jobs we are interested in running are to be found in the resources folder.

The jobs will read data (random and sequential) from your storage unit and print out a test report.

These changes has to be done in both files:

```
runtime = 120
direct = 1
```

The runtime changes will make the tools run 120 seconds and the direct changes avoid reading to memory.

Job files can be run with the following command `$ fio [FILENAME]`

The speed of a storage unit can be calculated by using the number of IOPS and the block size: IOPS * block size = speed (MiB/s).

For more about the output of the fio command, https://tobert.github.io/post/2014-04-17-fio-output-explained.html

Your tasks:

- Run `fio` with the job “fio-rand-read.fio”
- Run fio with the job “fio-seq-read.fio”
- Examine the results and write down the average number of IOPS.
- Calculate the read speed of the storage unit.

The exercise will return the following message:

```
Jobs: 1 (f=1): [R(1)][100.0%][r=1861MiB/s,w=0KiB/s][r=7445,w=0 IOPS][eta
00m:00s]
file1: (groupid=0, jobs=1): err= 0: pid=21: Wed Oct 30 13:26:52 2019
   read: IOPS=8499, BW=2125MiB/s (2228MB/s)(249GiB/120001msec)
    slat (usec): min=8, max=6699, avg=91.93, stdev=61.01
    clat (usec): min=143, max=92916, avg=1788.82, stdev=747.25
     lat (usec): min=277, max=93071, avg=1881.06, stdev=742.01
    clat percentiles (usec):
     |  1.00th=[  766],  5.00th=[  906], 10.00th=[ 1020], 20.00th=[ 1205],
     | 30.00th=[ 1385], 40.00th=[ 1565], 50.00th=[ 1729], 60.00th=[ 1893],
     | 70.00th=[ 2057], 80.00th=[ 2278], 90.00th=[ 2606], 95.00th=[ 2900],
     | 99.00th=[ 3654], 99.50th=[ 4113], 99.90th=[ 5473], 99.95th=[ 6521],
     | 99.99th=[10552]
    bw (  MiB/s): min= 1203, max= 2438, per=99.91%, avg=2122.90, stdev=243.65,
samples=239
   iops        : min= 4815, max= 9754, avg=8491.52, stdev=974.59, samples=239
  lat (usec)   : 250=0.01%, 500=0.01%, 750=0.74%, 1000=8.29%
  lat (msec)   : 2=57.56%, 4=32.82%, 10=0.58%, 20=0.01%, 100=0.01%
  cpu          : usr=2.76%, sys=79.78%, ctx=69600, majf=1, minf=1035
  IO depths    : 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=100.0%, 32=0.0%, >=64=0.0%
     submit    : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
     complete  : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.1%, 32=0.0%, 64=0.0%, >=64=0.0%
     issued rwt: total=1019915,0,0, short=0,0,0, dropped=0,0,0
     latency   : target=0, window=0, percentile=100.00%, depth=16

Run status group 0 (all jobs):
   READ: bw=2125MiB/s (2228MB/s), 2125MiB/s-2125MiB/s (2228MB/s-2228MB/s),
io=249GiB (267GB), run=120001-120001msec
```

```
Disk stats (read/write):
  sda: ios=2039907/21, merge=581/10, ticks=2162340/0, in_queue=2161920,
util=100.00%
```

Exercise 02*

In this exercise the student is going to run a write performance test of their drive, using the tool `fio`.

The setting is exactly the same as the previous exercise. The only difference is that these job files write instead of read. The job can be found in the resources folder.

The jobs will write data (random and sequential) to your storage unit and print out a test report.

These changes has to be done in both files:

```
bs = 4K
numjob = 4 (in rand)
numjob = 1 (in seq)
```

Your tasks:

- Run `fio` with the job “fio-rand-write.fio”
- Run fio with the job “fio-seq-write.fio”
- Examine the results and write down the average number of IOPS.
- Calculate the write speed of the storage unit.

The exercise will return the following message:

```
Jobs: 1 (f=1): [R(1)][100.0%][r=1861MiB/s,w=0KiB/s][r=7445,w=0 IOPS][eta
00m:00s]
file1: (groupid=0, jobs=1): err= 0: pid=21: Wed Oct 30 13:26:52 2019
   read: IOPS=8499, BW=2125MiB/s (2228MB/s)(249GiB/120001msec)
    slat (usec): min=8, max=6699, avg=91.93, stdev=61.01
    clat (usec): min=143, max=92916, avg=1788.82, stdev=747.25
     lat (usec): min=277, max=93071, avg=1881.06, stdev=742.01
    clat percentiles (usec):
     |  1.00th=[  766],  5.00th=[  906], 10.00th=[ 1020], 20.00th=[ 1205],
     | 30.00th=[ 1385], 40.00th=[ 1565], 50.00th=[ 1729], 60.00th=[ 1893],
     | 70.00th=[ 2057], 80.00th=[ 2278], 90.00th=[ 2606], 95.00th=[ 2900],
     | 99.00th=[ 3654], 99.50th=[ 4113], 99.90th=[ 5473], 99.95th=[ 6521],
     | 99.99th=[10552]
   bw (  MiB/s): min= 1203, max= 2438, per=99.91%, avg=2122.90, stdev=243.65,
samples=239
   iops        : min= 4815, max= 9754, avg=8491.52, stdev=974.59, samples=239
```

```
  lat (usec)   : 250=0.01%, 500=0.01%, 750=0.74%, 1000=8.29%
  lat (msec)   : 2=57.56%, 4=32.82%, 10=0.58%, 20=0.01%, 100=0.01%
  cpu          : usr=2.76%, sys=79.78%, ctx=69600, majf=1, minf=1035
  IO depths    : 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=100.0%, 32=0.0%, >=64=0.0%
     submit    : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
     complete  : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.1%, 32=0.0%, 64=0.0%, >=64=0.0%
     issued rwt: total=1019915,0,0, short=0,0,0, dropped=0,0,0
     latency   : target=0, window=0, percentile=100.00%, depth=16

Run status group 0 (all jobs):
   READ: bw=2125MiB/s (2228MB/s), 2125MiB/s-2125MiB/s (2228MB/s-2228MB/s),
io=249GiB (267GB), run=120001-120001msec

Disk stats (read/write):
  sda: ios=2039907/21, merge=581/10, ticks=2162340/0, in_queue=2161920,
util=100.00%
```

Exercise 03

In this exercise the student is going to implement a filesystem that uses MySQL for storage.

The resources for this exercise can be found in the resource folder.

This exercise does include nearly no help and relies on your own ability to solve the problems.

Code explanation

The methods `getaddr()` and `readdir()` are run when `$ ls -l` is executed.

`readdir()` returns a list of files.
`getaddr()` returns metadata about the file requested in path.
`read()` is run when the content of a file is read.

You can use the `encoded()` method in the `read()` method to encode the file content to UTF-8.

Helpful hints

- Install `python3`, `python3-pip` and `python3-fuse`

- Install `mysql.connector` and `fusepy` through pip: `$ pip install fusepy mysql-connector-python`
- Create a MYSQL database with a table (`Files`) containing the following columns:
  - `fileid (INT, auto_increment)`
  - `filename (TEXT OR VARCHAR)`
  - `filesize (INT)`
  - `filecontent (TEXT OR VARCHAR)`
- `$ ls -l` will list all files in the directory and `$ cat *` will read the content of files in the directory.
- If the solution is run in a Docker Container, use `--privileged`
- TODO comments in the code shows where you are supposed to include code.
- Help for using mysql in python: https://www.w3schools.com/python/python_mysql_getstarted.asp

Your tasks:

1. Make the filesystem work.